FOI-R-1129--SE
February 2004
ISSN 1650-1942

SEECS

FOI
SWEDISH DEFENCE
RESEARCH AGENCY

PROCEEDINGS

of the Second Swedish-American Workshop on Modeling and Simulation
SAWMAS-2004
Cocoa Beach, FL, USA, February 2-3, 2004

Editors: Johan Jenvald and Sören Palmgren

INSTITUTE OF TECHNOLOGY
LINKÖPINGS UNIVERSITET

INSTITUTE for
SIMULATION
& TRAINING

School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816-2362, USA

Division of Command and Control Systems
Swedish Defence Research Agency
P.O.Box 1165, SE-58111 Linköping, Sweden

PROCEEDINGS

of the Second Swedish-American Workshop on Modeling and Simulation SAWMAS-2004 Cocoa Beach, FL, USA, February 2-3, 2004

Editors: Johan Jenvald and Sören Palmgren

Computer and Information Science Department
Linköping University
SE-58183 Linköping, Sweden

Institute for Simulation and Training - UCF
3280 Progress Dr, Orlando, FL 32826, USA

Preface

Research on modeling, simulation and analysis (M&S&A) for applications to complex areas are nowadays taking place in virtually every research organization in society. Such complex applications include total defense planning; training and exercises in military and total defense environments; support to operations; technology development; and armaments acquisition. These circumstances have given rise to a multitude of networks on M&S&A with large differences in size and purpose.

This workshop originally intended to convene researchers in modeling and simulation from the United States and Sweden to exchange ideas and to form collaborative teams. However, it is with great pleasure the Organizing Committee can acknowledge the extended participation of individuals from countries outside that of the initiators. The Organizing Committee welcomes even further extension of the horizon of the Workshop in years to come.

The First Swedish-American Workshop on Modeling and Simulation, SAWMAS-2002 took place in Orlando, FL on October 30 and 31, 2002 and was hosted by University of Central Florida (UCF).

The Second Swedish-American Workshop on Modeling and Simulation, SAWMAS-2004 take place in Cocoa Beach, FL on February 2 and 3, 2004 and is hosted by University of Central Florida (UCF).

The Workshop is also an activity within the framework of the IEA-A-01-SW-1605 (Information Exchange Agreement) between Sweden and the USA in the area of Modeling and Simulation.

SAWMAS-2004 is supported by The School of Electrical Engineering and Computer Science at UCF; Institute for Simulation and Training – UCF; The Division of Command and Control Systems at the Swedish Defence Research Agency; The U.S. Army Research, Development & Engineering Command; and The Computer and Information Science Department at Linköping University.

Workshop Organizing Committee 2004

Dr. Avelino J. Gonzalez, SEECS-UCF, USA
Dr. Johan Jenvald, Visuell Systemteknik AB and Linköping University, Sweden
Dr. Sören Palmgren, FOI, Sweden

Honorary General Chair
of the Swedish-American Workshop on Modeling and Simulation

Dr. Erol Gelenbe

SAWMAS-2004 Program Committee

Dr. Clint Bowers, Team Performance Lab, UCF, USA

Dr. Mona Crissey, U.S. Army PEO STRI, USA

Dr. Henrik Eriksson, Linköping University, Sweden

Dr. Jennifer Fowlkes, CHI Systems Inc., USA

Dr. Avelino J. Gonzalez, SEECS-UCF, USA

Dr. Sture Hägglund, Linköping University, Sweden

Dr. Johan Jenvald, Visuell Systemteknik i Linköping AB, Sweden

Dr. J. Peter Kincaid, IST-UCF, USA

Dr. Magnus Morin, Visuell Systemteknik i Linköping AB, Sweden

Dr. Sören Palmgren, Swedish Defence Research Agency, FOI, Sweden

Dr. Judith S. Riess, Education & Training Solutions, USA

Dr. Alma M. Schaafstal, TNO Human Factors, The Netherlands

Dr. Greg Schow, U.S. Army RDECOM, USA

Dr. Ulf Söderman, Swedish Defence Research Agency, FOI, Sweden

Dr. Erland Svensson, FOI, Sweden

Table of Contents

Modeling and Simulation for Command and Control

Digitally Enhanced Situation Awareness: As an Aid to Battlefield Decision-Making

John Holmquist and John Barnett

US Army Research Institute of Behavioral and Social Science
12350Research Parkway
Orlando, FL 32826-3276
John.holmquist@ari.army.mil

Abstract

In combat and tactical situations, situation awareness is a key factor in the quality of decision-making. Currently, the US Army is investigating using digital systems, such as computer networks, digital imagery, and GPS, to enhance situation awareness at all levels of command. This article illustrates how digital technology is currently being used to enhance decision-making at the unit level and provides suggestions for how further advancements can be made. Further discussion is giving on the advantages of digital and embedded training in these areas.

Introduction

The military follows a chain of command in combat and decision-making. The higher echelons make broad decisions and each echelon below further refines the orders to address their part of the overall objective. In combat, situation awareness (SA) is a key factor in the quality of the decisions which are used to direct military forces. Any method that increases the commander's situation awareness also increases the effectiveness of the military unit.

Billings (1997) argues that the purpose of automation in aviation is to increase the aircrew's situation awareness. This is the same function that digitization, a form of automation, provides for the military decision-makers. Digital systems not only enhance situation awareness, but also facilitate shared situational awareness, which translates to a clear, accurate, and common picture of the battlespace for leaders at all levels.

In the past, scouts, observers on high ground, and even balloons have been employed to acquire information needed by commanders. Once acquired, this information must be transmitted to commanders by the best means currently available. In the early twentieth-century, information was communicated by signalmen and Morse code. World War II brought the radio to speed information back to the decision-makers.

Current technology has brought advanced sensors and satellite communications to the battlefield, bringing situation awareness to the decision makers faster and with more detail than ever before. Now, the key to decision-making is not only getting the information, but also filtering and using the information effectively.

Situation awareness is key to making decisions on the modern battlefield. To understand how digital systems support the commander's SA, it is first important to review situation awareness, battlefield decision making, and how the US Army uses digital systems.

Furthermore training of these types of technologies, by their nature as embedded equipment, support embedded training. An embedded trainer can be defined as a capability resident within the operational equipment that allows the trainees to use the operational equipment to receive training and practice skills (Andrews1991). Therefore skill acquisition will be quick and adequate for functional use.

Background

Throughout history, commanders have understood the importance of battlefield situation awareness in decision-making. Military operations are frequently shrouded by uncertainty (Von Clauswitz, 1984). This uncertainty has often been described as the "fog of war." The objective of obtaining tactical information was, and still is, to reduce this uncertainty. In a sense, situation awareness can be seen as the opposite of, or the counter to, the fog of war.

Situation Awareness

Situation awareness has been examined by a number of researchers, including Endsley, Fracker, and Harman and Secrist (Garland, Phillips, Tilden, and Wise, 1991). Randel, Pugh, and Reed (1996) studied situation awareness in the context of naturalistic decision-making. They found the ability to make better decisions was based, in part, on better situation awareness. In a similar study, Kaempf, Klein, Thordsen, and Wolf (1996) investigated how SA influences decision-making in a Navy Combat Information Center (CIC) and found that SA is an important factor in decision quality.

Endsley has developed a model of situation awareness. She suggests information about the environment is brought by the senses into the working memory, where it is formulated into a mental model of the outside world. Her model of situation awareness includes three levels: 1) a perception of the elements of the current situation, 2) comprehension of the situation, and 3) projection of future status (Endsley, 1995).

The US Army sees situation awareness in a much simpler form. SA is seen as the commander's mental model of the battlefield. Most often the term is used to describe information available on Tactical Operations Center (TOC) displays which allow commanders to develop this mental model. Such displays are often referred to as "situation awareness displays." The purpose of such displays is to provide decision-makers with enough information to make quality decisions.

Battlefield Decision-Making

The US Army has a formal process for planning military operations called the Military Decision-Making Process (MDMP) (Department of the Army, 1997). This process is not decision-making in the cognitive sense, rather it is more akin to structured problem solving.

Once plans are executed, decision-making becomes more cognitive and often follows the Recognition Primed Decision-Making model described by Klein (Adelman, Leedom, Murphy & Killam, 1998) where knowledge from the decision-maker's past experience is used as a framework for decisions about the current situation. Decision-making while units are in combat is characterized by the requirement to make decisions quickly. Often, commanders are bombarded with large amounts of information in various forms and must attempt to form a mental model of the situation to use as a basis for decisions. The limits of working memory preclude decision-makers from considering much of the information available (Wickens, 1992).

Digitization

The US Army digitization program seeks to capitalize on networked computer systems to enhance information flow and display. Theoretically, this allows decision-makers to maintain a clear and accurate vision of the battlefield necessary to support both planning and execution (Program Executive Office for Command, Control, and Communications, 2001).

In an Army unit equipped with digital systems, information is typically stored in common databases and can be accessed through a tactical internet, much like the global Internet. Much of the information, such as unit positions, can be displayed spatially as graphics, which is much easier to process cognitively.

The hardware of digitization presently consists of a family of systems. Each staff element, Intelligence, Artillery, Logistics, Air Defense, and Maneuver, has a computer system specifically designed to meet its unique needs. However, this entire family of systems is networked together to share information. The final member of this family is a computer system which is customized for lower echelon units, the troops in the field. This system, known as the Force XXI Battle Command, Brigade and Below (FBCB2), allows access to the information network down to individual fighting vehicles, such as tanks and infantry fighting vehicles.

Digital Systems To Enhance SA: Current Methods

This digital network allows commanders to maintain an awareness of their subordinate units, known as friendly SA. In mechanized units consisting of tanks and infantry fighting vehicles, for example, each vehicle tracks its geographical position by means of a Global Positioning System (GPS) receiver. Periodically, its position is transmitted back to the unit network where it can be displayed on the commander's computer. This ensures the commander knows the location all the unit's vehicles at all times. A study conducted by McGuinness and Foy (2000) found commanders rated this factor to be one of the most helpful.

As well as friendly SA, commanders also require SA concerning the enemy. Military Intelligence specialists access advanced sensors to locate enemy units and enter information into the database. Once the location of enemy units on the battlefield can be accurately presented, commanders can recognize patterns of activity and estimate the enemy's intent. With this information, the commander's options will become more clear.

Similarly, each individual vehicle equipped with FBCB2 can access the same information; therefore, soldiers in the field are aware of the location of other vehicles in the unit in relation to enemy units. Because the digital computer systems are in the hands of leaders and soldiers alike, digitization provides not only improved situational awareness but shared SA. This timely sharing of information allows better coordination among units and significantly improves the ability of commanders and leaders to make decisions quickly. (Department of Defense, 2000).

Future Possibilities

Through the application of advanced technology on the battlefield, the Army is well on its way to establishing full situational awareness (Department of Defense, 2000). The problem with displays to enhance situation awareness is not just presenting the information, but presenting it in such a way that it can be processed by the decision-maker into an accurate mental model of the situation. Simply presenting more information does not ensure better SA. In fact, presenting too much information can overload the recipient and actually inhibit SA. Displays that take advantage of an understanding of human information processing could present more useful information to the decision-maker.

Commander's HUD

Considerable effort has been expended producing Head Up Displays (HUD) in fighter aircraft to enhance the pilot's SA. The essence of the HUD is to present real-time information in the center, or focal, part of the display and include status or secondary information around the periphery. This allows pilots to focus attention on the most important information in the center of the display, yet perceive secondary information while minimizing the attentional and perceptual cost of accessing the information.

The same technology might be used to provide see-through displays for lower echelon commanders, such as tank or infantry fighting vehicle commanders. Computer generated graphics might be projected onto vision blocks and external sights to present important mission-oriented information such as navigational way points, suspected targets, or danger areas.

For example, as a vehicle approaches an area known to be mined, an image could be projected into the vision blocks so that the area appears to glow red, an indication the area is dangerous and should be avoided. Or, since the positions of friendly forces are known, symbols for friendly vehicles could be displayed in the tank gunner's sight to help the gunner distinguish between friendly vehicles and enemy targets.

Minimizing Information Access Cost

One of the difficulties in designing head up displays is that the information presented in the periphery should be in a format which is easy to perceive and process. Typically, such information is presented in symbolic or spatial formats. For example, a display in a helicopter may present the aircraft's altitude above the ground as a bar on the side of the display that rises and falls as altitude increases and decreases. This symbology is easier to process cognitively than forcing pilots to convert numbers to height above ground.

Similar modeling could be used for digital displays to allow commanders to more easily process information. Minimizing information access cost, that is, reducing the amount of cognitive effort required to perceive information, is an important consideration in display design (Wickens, Gordon, & Liu, 1999). Suppose a commander had a geographical display that showed his forces in spatial relationship to enemy forces and significant terrain features. Peripheral information could also be displayed on the edges of the display, provided they had the appropriate perceptual coding.

For example, the status of each of the commander's subordinate units could be displayed as color-coded symbols. Each unit might have a symbol for fuel status, ammunition status, and personnel status, coded green if the status was acceptable, yellow for marginal, and red for unacceptable. The status of subordinate units would be readily apparent. Any symbol that was not green would focus the commander's attention on that unit. The commander could then access more specific information on the unit's status by selecting it with a cursor. Such a status display would enhance the commander's SA of unit status. Using color-coded symbols would reduce cognitive information processing costs.

Currency of Information

Other techniques could be used to project information at low cognitive loading. For example, one important consideration of situational displays is how current the information is. Icons that symbolize a moving unit may not be updated on a regular basis. Therefore, the unit may not be where it is depicted by the symbol. If it is not possible to update the symbol in real time, the viewer should be able to tell how long it has been since the last update.

A technique used in some computer games is to have the symbol "fade" out as information ages. This fade-out is accomplished by changing the colors from bright to dim over time. Such fading takes advantage of user experiences in the world, a technique known as mapping (Norman, 1988). Many physical objects fade in color as they age. The fading of the symbols maps the fading of objects as they get older. This type of natural mapping uses minimal attentional resources to process the information.

Predictive Aiding

Wickens, Gordon, & Liu (1999) suggest that since predicting future events is a difficult cognitive task, it is an ideal task to delegate to automated displays. Prediction often requires complex mathematical calculations, which, although difficult for people, present little problem to computer systems. Predictive aiding could support situation awareness by helping the commander to visualize the relative positions of friendly and enemy forces at a future time.

Predicting where moving units will be in the future, how much fuel they will use, and when critical events should occur are all well within the abilities of digital systems.

Training

With the progress in the development of techniques, such as user modelling and capture of real-time operator data, the potential for an automated comparison of operator performance against training objectives in real time, and the provision of on-line feedback, or remedial instruction to the trainee is available for these skills (Zachary, Ryder, Hicinbothom & Bracken, 1997).

Conclusions

Generally speaking, the better information a decision-maker has, the better decisions are made. The use of digital automated systems to increase situation awareness is a promising method to allow decision-makers to develop a more accurate mental model of the situation, and consequently increase the quality of decisions.

This high level of information gathering and sharing is not limited to military applications. In the civilian world, digital enhancements for SA can be used to help guide drivers through congestion areas, monitor crowd movement at major events, such as sports or political gatherings, and better match public transportation to the changing needs of its patrons.

The ideas listed in this paper are merely examples of how electronic information processing and display tools can help enhance situation awareness and support decision-making. The intelligent application of human factors to systems design should provide a wide range of technological aids to support human information processing and decision-making.

References

Adelman, L., Leedom, D. K., Murphy, J. & Killam, B. (1998). *Description of Brigade C2 Decision Process* (Army Research Laboratory Technical Report). Aberdeen Proving Grounds, MD: Army Research Institute.

Andrews, D. (1991) Embedded training and embedded practice. Educational technology, 21 (10),41-44

Billings, C. E. (1997) *Aviation automation: The search for a human-centered approach.* Mahwah, NJ: Lawrence Erlbaum Associates.

Department of the Army (1997). *Field Manual 101-5, Staff Organization and Operations*. Washington, DC: Author.

*Department of Defense (2000, April) *Report on the Plan for Fielding the First Digitized Division and First Digitized Corps*: Presented to the Committee on Army Services, United States Senate, Second Session, 106th Congress Washington: DC. Author.

*Endsley, M. R. (1995). Theoretical underpinnings of situation awareness: A critical review. *Proceedings of the International Conference on Experimental Analysis and Measurement of Situation Awareness, Daytona Beach, FL*, (pp. ?-?). November 1-3. Daytona Beach: Embry-Riddle Aeronautical University.

Garland, D. J., Phillips, J. N., Tilden, D. S. & Wise, J. A. (1991). *Theoretical underpinnings of situation awareness: Towards an objective measure.* (Tech. Rep. No. CAAR-15498-91-1). Daytona Beach, FL: Embry-Riddle Aeronautical University, Center for Aviation/Aerospace Research.

Kaempf, George L., Klein, Gary, Thordsen, Marvin L., Wolf, Steve (1996). Decision making in complex naval command and control environments. *Human Factors 38*(2) 220-231.

*McGuinness, B & Foy, L. (2000). Battlespace digitization: SA issues for commanders. *Proceedings of the Human Performance, Situation Awareness, and Automation Conference* , pp. 125(-?. Dates). Marietta, GA: SA Technologies.

Norman, D. A. (1988) *The Psychology of Everyday Things*. New York: Basic Books.

*Program Executive Office Command, Control and Communications Systems (2001). *Our Leadership in Digitization*. [Online] Available: http://peoc3s1.monmouth.army.mil/battlefield_digitization.htm

Randel, J. M., Pugh, H. L. , Reed, S. K. (1996). Differences in expert and novice situation awareness in naturalistic decision making. *International Journal of Human Computer Studies, 45*(5) 579-597.

Salas, E., Prince, C., Baker, D. P. & Shrestha, L. (1995). Situation awareness in team performance: Implications for measurement and training. Special Issue: Situation awareness. *Human Factors, 37*(1) 123-136.

Sarter, N. B. & Woods, D. D. (1995). How in the world did we ever get into that mode? Mode error and awareness in supervisory control. Special Issue: Situation awareness. *Human Factors, 37*(1) 5-19.

Von Clausewitz, C. (1984). *On war*. (M. Howard and P. Paret, Eds., Trans.) Princeton, N.J.: Princeton University Press. (Original work published 1820)

Wickens, C. D. (1992). *Engineering Psychology and Human Performance* (2nd Ed.). New York: Harper Collins.

Wickens, C. D., Gordon, S. E. & Liu, Y. (1998). *An Introduction to Human Factors Engineering*. New York: Longman.

Zachary, W., Ryder, J., Hicinbothom, J., & Bracken, K (1997) The use of executable cognitive models in simulation-based intelligent embedded training. *Proceedings of the HFWS 41st Annual Meeting*, pp. 1118-1122.

A Research Agenda for Critiquing in Military Decision-Making

Ola Leifler   Henrik Eriksson
Dept. of Computer and Information Science
Linköping University
SE-581 83 Linköping
Sweden
{olale,her}@ida.liu.se

Abstract

Supporting military decision makers is an important but difficult task. Providing correct and meaningful feedback on user input during the military planning process is especially difficult, given the characteristics of modern staff environments. In such environments, much work is performed simultaneously by domain experts who need to coordinate their efforts in joint operations. The products of the planning process, the orders, take the form of rough guidelines to subordinate units and are not intended for computer interpretation. We argue that there is a need for new approaches to much of what has traditionally been seen as decision support. In this paper, we discuss these difficulties in detail and outline a research agenda to address the most pressing issues at hand for decision support systems.

Introduction

Military decision-making is an activity where the conditions for making decisions are extreme—human lives depend on the decisions made and the time and information constraints can make the task of a military officer daunting. Much effort has therefore been put into trying to aid military decision-makers at all levels to improve the conditions under which they make these decisions (see Section Expert Critiquing systems).

Because it is not acceptable or desirable to automate the process of making military decisions, researchers have investigated how the individual decision-makers can be supported by the use of modern technology. Out of the issues in decision-making that are best suited for the use of technology, information overload and communication problems have been well addressed by researchers in the field of Computer Supported Collaborative Work, CSCW [Schmidt and Bannon, 1992]. During the last decade, the very decision-making process has received more attention as techniques from the area of Artificial Intelligence and, in particular, Expert Systems have gained popularity as a means of improving the interactions with computer systems that provide decision support.

Traditionally, research on Expert Systems and the related Critiquing Systems have focussed on scenarios where a computer acts as an advisor that responds to user input in an intelligent way, either by presenting suggested solutions to problems that the user presents or by analyzing a given solution according to some rules that are provided by a human subject matter expert. For successful deployment of decision-support systems in a contemporary staff environment, use scenarios that involve a group of decision-makers must be considered for any system. Therefore, we believe that there is a need for research on collaboration aspects of decision support.

One of the future challenges for this area of research is to find suitable designs for decision-support systems that can function in an operative staff environment; that is, in a *group* of people collaborating on the task of planning and executing military operations. The additional complications in developing such systems, compared to constructing traditional decision support, stem from both the need of synchronizing a large group of people and also from the need of the computer system having a coherent view of the current state of the planning process.

The aim of this paper is to identify and present future directions for research in critiquing systems that specifically address the need to support *groups* of decision makers. First, we describe the environment of the conceptual Command and Control system ROLF 2010 in Section Background. We proceed by describing some contemporary military critiquing systems in Section Expert Critiquing systems. As we believe that the cooperative aspect of decision support is one of the more interesting future directions for such systems, we outline what could be done in terms of supporting the group in making decisions in Section Cooperative work. Finally, we discuss the potential benefits and drawbacks of our proposal in Sections Discussion and Final remarks.

Background

ROLF 2010 [Sundin and Friman, 2000] is a research project on future Command and Control (C^2) Systems conducted by the Swedish National Defense College. The overall vision of the project is to develop a staff environment where a small group of military officers in an operative staff (5–10 officers as compared to several hundreds today) conduct the day-to-day routines of an operative staff using advanced computer support. This support should consist of the fusion of sensor data from intelligence units, distribution of relevant information (orders and briefs) to other units and *intelligent user support* that help the officers in making their decisions.

Since 1995, there has been much work in the areas of collaboration studies [Granlund et al., 2001, Artman and Persson, 2000], visualization studies [Persson, 2000], and simulation tools [Woodcock et al., 2003], which cover the first two kinds of computer support listed above. In this paper, we shall focus on giving an outline for the third part of the project—intelligent decision support.

ROLF 2010 has been devised as a consequence of the Swedish Governments agenda to reshape the Swedish Defense into a "Network Organization"—an organization with less rigid structure of command. It has been argued that Network Organizations would be better suited to respond to changes in the environment than traditional, hierarchical ones and that many of the problems of traditional C^2 systems could be solved by using this new structure [Alberts et al., 2000, Cebrowski and Garstka, 1998, Sundin and Friman, 2000]. One of the key features of this new organization would be that fewer levels of control are needed to perform the same tasks as before. Each level of control is believed to add to the rigidity of an organization and prolong the reaction time. However, one of the problems with this Network Organization is that commanders at all levels must process more information in order to exert efficient command and control over their units due to the delegation of responsibilities. Processing more information in stressful circumstances makes computer support even more important, both in presenting an overview of the current situation but also for communicating efficiently with other commanders.

Expert Critiquing systems

Silverman defined the concept of a critiquing system as follows [Silverman, 1992, p. 4]

> The main tasks of critics are [...] threefold: (1) to recognize and analyze human error; (2) to form persuasive criticisms and feedback; and (3) to adapt to the situation and any lessons learned.

This definition may not be very useful when trying to understand what critiquing systems have really been like—what does it mean to "analyze human error"? Are only erroneous behaviours the ones a critiquing system should give feedback on? In Section Redefining Critiquing we propose an alternative to traditional critiquing systems where feedback on erroneous behaviour is not the important part. However, adaptation (the third part of Silvermans definition) remains vital in our proposal and we do believe that our ideas fall within the domain of critiquing.

So what have traditional critiquing systems been like? Traditionally, critiquing systems have consisted of propositional logic formulas that describe rules that are valid in a certain domain of interest. These logical rules are provided by a human domain expert and express consequence relations on the form `IF condition THEN consequence` where both `condition` and `consequence` are in themselves logical expressions. A user of the critiquing system proposes a "solution" to some "problem" in the domain and gets feedback on his "solution" based on these rules, if necessary.

As an example, consider the domain of moving military units and a critiquing system that is supposed to help an officer decide on appropriate ways of moving a set of units to a given location on the map. The rules that an expert could provide the system with for this scenario could be on the form

```
IF class(unit)=tank AND
   terrain(location(unit))=forest
THEN
   defensive_value(unit)=low
```

This logical rule expresses the fact that if a tank is placed in a forest it will have poor defensive capabilities. A user that proposes to move his tank units through forestrial terrain as a part of his plan could then receive a warning if the system is instructed to warn whenever the `defensive_value` function returns the value `low` for some unit during the movement process.

Note that there may be different representations of knowledge that is used in a logical inference system. The concept of "logical rules" is clearer when using the representation above though.

These kinds of systems have been studied and deployed in medical environments [Shahar et al., 1998] and recently, in military settings [Gil et al., 2003]. In the following sections, we present an overview of some of the projects that have been conducted on supporting military decision-makers. These projects have done research on several aspects of critiquing systems as described above that are particular to a military staff environment, for example

1. interfaces for creating computer representations of military plans,
2. interfaces for creating logical rules by using graphical representations,
3. methods for verifying the correctness of rules,
4. methods for using rules to give feedback on the users' solutions, and
5. design of multi-agent systems where the user interacts with several critiquing agents and cooperates with them to find the solution to the problem at hand.

When talking about what has been studied regarding these aspects, we shall use SHAKEN [Gil et al., 2003], CADET [Kott et al., 2002, Group, 2003], SHAKER/KRAKER [Pool et al., 2003], ArCS [Li et al., 2000], Disciple [Learning Agents Lab, 2003], MRE (Mission Rehearsal Exercise) [ICT, 2003, Traum et al., 2003] and EXPECT [Valente et al., 1996] as examples of projects. DARPA funded all these projects through the programmes High Performance Knowledge Bases [Gunning, 2003] and Rapid Knowledge Formation [DARPA, 2003].

Representing military plans

CADET was a project aiming at creating military plans according to the "Course of Action" formalism [US Army, 1997] by inferring as much information as possible automatically from the users' description of an intention.

The product from the CADET tool was a description of the actions necessary to achieve a tactical goal and with a synchronization matrix that described all the ordering constraints inferred between the actions in the plan.

ArCS on the other hand focused on how to use visual representations of military units to deduce logical representations of the state of units on the map (for example locations and movements). The ambition was to deduce, from the map and the symbols on it, information like whether a flank attack was being planned by the user or if units moved on trafficable terrain. Such qualitative information proved hard to formally describe and deduce correctly and was one of the major challenges with the project.

Creating and verifying rules

SHAKEN was an attempt to let domain experts themselves enter *and verify* logical rules describing features of a military scenario. The experts used two methods to verify their rules—in a declarative inference engine where rules describing trafficability of terrain are used to control whether legal actions are performed by the user and simulation in a Capture the Flag scenario where the rules are used to evaluate the performance over time as the plan unfolds. Synchronization problems were more easily presented to the experts in such a simulation compared to using declarative inference.

Disciple tried the "teaching" approach, where an expert would instruct a software "agent" to recognize a Center of Gravity in scenarios that was use in a military course. The agents ask, through spoken dialogue, the instructor for information on how to recognize the center of gravity of the enemy.

Using rules

The synchronization matrix of the CADET tool was used as a basis for a simulation in which it determined the quality of the users' actions by having an opponent with some AI that would react to the proposed plan and thus simulate the dynamics of the real situation. Apart from using this matrix, CADET helped the user to create a plan by informing him of the available resources and other constraints. Though these constraints were known to the system, it did not force the user to follow the restrictions they set since during the planning process, much as a word processor would allow a user to construct grammatically incorrect sentences. For the user, it would be more intrusive and less efficient to require each sentence to be correct before proceeding with the next one.

INSPECT made grammatical verification of the plans mostly, controlling both domain independent and domain dependent properties in the plan structure. The users plan was a description of an Air Force mission in the form of a tree-shaped graph with a root node describing the overarching goal, and children nodes containing more specific information about aspects of the plan compared to the respective parents.

Domain-independent checking consisted of for example verifying that all nodes except the root node had children nodes. Domain-dependent checking was for example in the form of controlling that the information specified in each node was more specific than the parent's node.

Multi-agent systems

In MRE, several AI agents were to cooperate with a human operator in solving a problem. This was to be performed in an environment where the human was in charge of a small group of UN soldiers on a mission to help humanitarian transports into an area. The AI informed the operator of the situation and could also receive orders. As for critiquing, all agents were responsible for different parts of the operation and had opinions on what was important to do. Thus, the "critiquing" consisted mostly of a discussion between the agents and the human on how to proceed from a given situation to achieve some goal.

Cooperative work

In supporting the *collaboration* in a team of decision-makers, there could be many improvements made to current critiquing systems. In 1988, Kraemer and King [Kraemer and King, 1988] argued that none of the decision support systems of that time had been successful. This failure, they argued, was in part because the very concept of decision support was not well defined, but also because none of the systems analyzed by Kraemer and King had taken into account the specific needs for decision support that each potential user group might actually have.

Within the ROLF 2010 project, behavioristic studies [Artman and Persson, 2000] suggest that even though the staff environment has been designed specifically to support equal interaction and making all staff members each others peers, the military code of conduct and traditions make it difficult to achieve discussions on alternative plans. The question is whether a traditional approach to decision support, as described in Section Expert Critiquing systems, would offer relevant support for the ROLF 2010 environment.

There has been interesting research in the area of supporting groups that collaborate, although this has not primarily been aimed at supporting decision making. It is difficult to envision just how and why the computer support described in Section Expert Critiquing systems would be the most efficient way to support the group in their decision making. The group does not make their joint decisions by separately making plans that are in themselves complete and "correct" and later merging them into a coherent whole. Rather, the group assembles at times and works individually otherwise. During their work, orders of various specificity and duration are produced and handed out. Changes to standing orders are delivered more regularly than as an exception. That is, there is not a single irrevocable order that needs to be made perfect, nor is it easy to define when a system can access a final plan to analyze and criticize. However, during the planning process we believe there are many situations in which *feedback on user input* (not necessarily on human error as defined in the definition in Section Expert Critiquing systems) could be provided in a way that would efficiently support the collaboration.

Adaptable groupware

In projects related to the design and use of group-ware applications [Malone et al., 1995, Roseman and Greenberg, 1996], for example OVAL [Malone et al., 1995], researchers studied how end-users themselves could build all the basic functionalities of larger groupware applications. These results are interesting from the point of view that they highlight possibilities for constructing and customizing the applications that support a distributed group of people in need of synchronizing their activities.

When the different groups are working independently from each other, they still work with the same information, for example a list of targets or resources. Later, they meet to synchronize their efforts so that their joint effort does not consume resources that are not available and that their intentions do not clash or contradict the overall plan. However, if they only then discover that they have been talking past each other with respect to what resources should be used for which purposes, it would probably be more time-consuming than if there could be continuous feedback and possibility to monitor the effects on your own intentions by other staff members' plans.

For example, consider a joint UN operation where officers from several countries are to cooperate on helping convoys with medical and other supplies into a war zone during a cease fire in a conflict. These commanders handle different aspects of the operation, like maintaining road blocks that allow for the convoys to pass through, patrol the area to avoid ambushes and informing the locals what is going on during the operation. This operation needs to be expedient since the cease-fire is about to expire and hostilities are to be expected in the region when it does. Thus, the coordination of this effort is crucial and the commanders need to have constant information on the progress of the other commanders' activities. However, constantly meeting and briefing each other is inefficient and would require too much time. Thus, they would need to asynchronously inform each other of the progress made, preferably by having their activities logged and sending information to other unit commanders that is of importance to them.

Discussion

In this section, we shall describe and discuss some scenarios of how to use an *adaptable* critiquing system that is intended for group support primarily.

Redefining Critiquing

The essence of critiquing is to support the user by giving feedback on user input in an intelligent fashion, preferably by using expert knowledge of the domain in question. However, as the context becomes that of a group collaborating to solve a set of tasks, it is less obvious how this should work. Would the most efficient way be to let the group jointly present their decision to a computer system, which would only *then* evaluate the teams decision with respect to known agendas, expert domain knowledge and recommended practices? Would the feedback then come too late to be of any real value?

Alternatively, should the expert system be more seamlessly integrated in the work environment of the team, where software used for cooperative work could be enhanced to in itself support continuous feedback on the actions taken by the users?

Traditionally, the expert/critiquing systems have been organized so that first experts have entered their knowledge of how a system should react, with or without the help of knowledge engineers, and then the end-users, presumably non-experts, use this knowledge to compare their own decisions with. However, in a decision environment as in ROLF 2010, one can expect all of the users of a computer system to be seasoned experts in their field of expertise. Such an environment makes it difficult to develop an expert system since, apart from standing operating procedures, there is not much historical wisdom that can be taken for granted and used as common-sense knowledge that the users of the system are not already well aware of. However, since they hopefully know well how to perform their individual tasks, they probably do not have much problem formulating simple rules about cause and effect in military scenarios. Thus, one could imagine letting themselves have an adaptable environment for constructing rules that could be used to analyze data they work with, much as was done in the OVAL project [Malone et al., 1995].

Let us consider for a moment how this kind of critiquing framework could work and in which scenarios it would be useful.

Staff briefing Following the scenario description in Section Cooperative Work, one could imagine that the commander in charge of road-blocks need to know which route the convoy is going to take so that he can plan for his deployment and withdrawal of troops. Likewise, the information officer who is responsible for information the public about the operation needs to know at what times the UN convoys will enter the area so that those in need of supplies can leave their shelters. They are now in the process of planning the operation and are doing so in their respective command centers, though able to access the same information through a computer system. In order to handle this need for synchronization, the commanders construct rules by using graphical elements representing the common elements available during the operation where the possible preconditions and actions of rules are available[1]. These rules are then activated and they will result in notifications whenever some specified precondition holds.

Following the commanders intent When the staff members have decided on an agenda for the next period of days, they distribute this information to the units on the tactical level that are supposed to carry out the orders.

Since these units are supposed to be fairly autonomous and have the responsibility to solve their tasks only with the help of an overarching goal and some broader limitations (contained in the order they are given), it is imperative that the commanders intent with an operation is made clear to all parties involved. However, in order to focus on what is important for a specific unit and solve their individual missions as they see fit, at the tactical level you need to be able to extract only the information that is relevant and necessary. In doing so, there is the risk of losing focus on getting the whole operation to work, so the tactical commander as well as the operational staff would be greatly helped if the orders were delivered along with a set of rules generated from the order that could help in finding possible problems that arise during the operation.

In both these scenarios, the critiquing system is not used traditionally—that is, having a few experts feed domain information about solutions to common problems into a system that later provides critique to non-expert users.

Why constructing rules?

There could be many objections to the proposed framework for critiquing applications in a group context that we presented in Section Redefining Critiquing.

Why should high-ranking officers who have this important task of planning and monitoring military activities have to learn how to program a rule-based, distributed critiquing system in order to do their job? Discussing the benefits of end-user programming does not need to be limited to the context of military critiquing systems, but can be applied to whether or not to teach non-programmers in general more sophisticated uses ("programming") of a computer system.

[1] elements could in this scenario include convoys, whereas a possible precondition might consist of the convoy entering a certain area of the map and an action could be sending an email to the officer in charge of road blocks in that area

If the end-users, who are assumed to be non-programmers, need to have a very simplified language as their programming tool for tailoring the applications they handle, then there is a risk that the expressivity of the language becomes too limited to be of any real use. The range of rules that can be constructed might not cover the needs that arise in real-life situations. Because no experiments have been conducted using the ideas presented in Section Redefining Critiquing and no system for constructing such rules has been devised we cannot make any statements on whether overly complicated uses of rules would really be necessary in order to have some benefit of tailorability per se. However, in many application there has been no mention of programmability yet a range of modifications can be made and performing the modifications is quite similar to programming (for example, constructing email filtering rules). Such modifiability is typically appreciated by users.

We can imagine that if a group-oriented critiquing system becomes widely adopted, many unforeseen needs could arise among the users of the system. The ability to construct simple rules may prove to be reason enough for most potential users to use the system, though it could of course also be the case that the hard problems in monitoring the activities of other people is not something which can readily be solved by a computer system. If this is true, that is, if there is really no need for computer support when it comes to the synchronisation and monitoring of a group, then it would seem as if much effort had been put into achieving nothing. However, making such negative statements would require some underpinnings in the form of investigations on users needs. Such investigations would be natural to include as one of the parts of our proposed project.

Are people willing to use a system that they are required to modify in order to use efficiently? Studies on how people work with computer systems as a tool for supporting joint activities might be misleading. Those that are currently in charge of military operations are not often representative to younger commanders in their use of modern technology. Difficulties in using computer systems that current staff members might have may well be non-issues in five to ten years from now. We believe that younger commanders would show use patterns that are more advanced than their priors are and that they would be less averse against adapting computer systems for their own needs.

Would there then be anything new in letting officers construct “critiquing rules” much in the same way as a e-mail filter is customized and using those rules much the same way as post-it notes (assuming we use rules for notification only)? It is not intended that the examples in this paper should be taken too literally and interpreted as the only way we see adaptable decision-support mechanisms for groups of decision makers, but we believe there are elements in our proposal that are new compared to both traditional groupware applications and to traditional critiquing systems.

First, we do not primarily focus on constructing knowledge[2] *that can distinguish good decisions from bad ones*, but we rather see how the concept of intelligent feedback can be used to support a group of people in the process of making a decision. This feedback is not primarily given on decisions but on any action performed.

Second, we do not intend to reinvent concepts commonly used in CSCW systems, such as synchronized timetables or other already existing technologies for group collaboration. Rather, we wish to explore how the expert knowledge that the staff members have of each situation can be used to communicate the relevant information to each other and to have customized, and thereby more efficient, means of getting feedback on their planning and monitoring of operations.

[2] or supporting such construction

Final remarks

Future research on decision-support systems will need to focus on the work process of the group in order to support them efficiently. We do not believe that it is likely traditional critiquing systems will be the primary tool for helping in the collaboration of a military staff in the future.

Those in control of an operation are also the ones who know best what information they need to share with others and how they need to cooperate to reach their goals. Therefore, they should also be in control of the computer system they use for their support.

Acknowledgements

This work was supported by the Swedish National Defence College (FHS).

References

[Alberts et al., 2000] Alberts, D. S., Gartska, J. J., and Stein, F. P. (2000). *Network Centric Warefare: Developing and Leveraging Information Superiority*. National Defense University Press, Washington, DC.

[Artman and Persson, 2000] Artman, H. and Persson, M. (2000). Old practices – new technology: Observations of how established practices meet new technology. In *Designing Cooperative Systems – The Use of Theories and Models. Proceedings of the 5th International Conference on the Design of Cooperative Systems (COOP’2000)*, pages 35–49, Sophia-Antipolis, France.

[Cebrowski and Garstka, 1998] Cebrowski, A. K. and Garstka, J. J. (1998). Network-centric warfare: Its origin and future. *U.S.Naval Institute Proceedings*, 124(1):28–35.

[DARPA, 2003] DARPA (September, 2003). Darpa rapid knowledge formation programme. http://www.ksl.stanford.edu/projects/RKF/.

[Gil et al., 2003] Gil, Y. et al. (2003). A knowledge acquisition tool for course of action analysis. In *Proceedings of the 15th Annual Conference on Innovative Applications of Artificial Intelligence*, pages 43–50, Acapulco, Mexico.

[Granlund et al., 2001] Granlund, R., Johansson, B., and Persson, M. (2001). C^3Fire: a microworld for collaboration training in the ROLF environment. In *Proceedings of the 42nd Conference on Simulation and Modeling*, Porsgrunn, Norway.

[Group, 2003] Group, L. C. (2003). CADET. http://www.arl.army.mil/aro/arowash/rt/sbir/99brochure/carnegie.htm.

[Gunning, 2003] Gunning, D. (Sepmteber, 2003). High performance knowledge bases. http://reliant.teknowledge.com/HPKB/.

[ICT, 2003] ICT (Sepmteber, 2003). Mission rehearsal excercise. http://www.ict.usc.edu/disp.php?bd=proj_mre.

[Kott et al., 2002] Kott, A., Ground, L., Budd, R., Rebbapragada, L., and Langston, J. (2002). Toward practical knowledge-based tools for battle planning and scheduling. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pages 894–899, Edmonton, Alberta, Canada.

[Kraemer and King, 1988] Kraemer, K. L. and King, J. L. (1988). Computer-based systems for cooperative work and group decision making. *ACM Computing Surveys*, 20(2):115–146.

[Learning Agents Lab, 2003] Learning Agents Lab, G. (Sepmteber, 2003). Disciple. http://lalab.gmu.edu/Projects/disciple/disciple.htm.

[Li et al., 2000] Li, J., Condoravdi, C., and Pease, A. (2000). From visual to logical representation – a GIS-based sketching tool for reasoning about plans. Technical report, Teknowledge Inc.

[Malone et al., 1995] Malone, T. W., Lai, K.-Y., and Fry, C. (1995). Experiments with OVAL: A radically tailorable tool for cooperative work. *ACM Transactions on Information Systems*, 13(2):177–205.

[Persson, 2000] Persson, M. (2000). Visualization of information spaces for command and control. In *ROLF 2010 – The Way Ahead and The First Step*. Gotab Erlanders, Stockholm.

[Pool et al., 2003] Pool, M. et al. (2003). Evaluating SME-authored COA critiquing knowledge. In *Proceedings of K-CAP 03*.

[Roseman and Greenberg, 1996] Roseman, M. and Greenberg, S. (1996). Building real-time groupware with groupkit, a groupware toolkit. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 3(1):66–106.

[Schmidt and Bannon, 1992] Schmidt, K. and Bannon, L. (1992). Taking CSCW seriously. *Supporting Articulation Work, Computer Supported Cooperative Work – An International Journal*, 1(1–2):7–41.

[Shahar et al., 1998] Shahar, Y., Miksch, S., and Johnson, P. (1998). The Asgaard Project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines. *Artificial Intelligence in Medicine*, 14(1–2):29–51.

[Silverman, 1992] Silverman, B. G. (1992). *Critiquing Human Error – A Knowledge Based Human-Computer Collaboration Approach*. Academic Press, London.

[Sundin and Friman, 2000] Sundin, C. and Friman, H., editors (2000). *ROLF 2010 – The Way Ahead and The First Step*. Gotab Erlanders, Stockholm.

[Traum et al., 2003] Traum, D., Rickel, J., Gratch, J., and Marsella, S. (2003). Negotiations over tasks in hybrid human-agent teams for simulation-based training. In *Proceedings of the Second Joint Conference on Autonomous Agents and Multiagent Systems*, pages 441–448, Melbourne, Australia.

[US Army, 1997] US Army (1997). *Field Manual 101-5: Staff Organisation and Operations*. Department of the Army, Washington, D.C.

[Valente et al., 1996] Valente, A., Gil, Y., and Swartout, W. (1996). INSPECT: An intelligent system for air campaign plan evaluation based on EXPECT. Technical report, USC – Information Sciences Institute.

[Woodcock et al., 2003] Woodcock, A. E. R., Hitchins, D. K., and Cobb, L. (Sepmteber, 2003). The strategic management system (STRATMAS) and the deployment of adaptable battle staffs. http://www.dodccrp.org/Proceedings/DOCS/wcd00000/wcd0005c.htm.

Bandwidth Optimizations for Integrated Tactical and Training Networks

Carlos Leon-Barth, Ronald F. DeMara, Avelino J. Gonzalez, Michael Georgiopoulos
Intelligent Systems Laboratory School of Electrical Engineering and Computer Science University of Central Florida
Orlando, FL 32816-2450 jcleonbarth@hotmail.com, rdemara@mail.ucf.edu, gonzalez@pegasus.cc.ucf.edu, michaelg@pegasus.cc.ucf.edu

Abstract

This paper addresses the bandwidth and latency optimization of Embedded Simulation (ES) communications within tactical Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) networks while supporting an Enroute Mission Planning and Rehearsal (EMPR) for ground combat vehicles and other use cases. Simulation data obtained from One Semi Automated Forces (OneSAF) Testbed Baseline simulations is consistent with Future Combat Systems (FCS) Operations and Organizations scenarios of multiple-platoon, company, and battalion-scale force-on-force EMPR vignettes. The resultant simulation traffic is modeled and assessed within a hierarchical communication architecture consisting of Manned Platforms, Distributed Common Ground Systems (DCGS_A) and Multiband Integrated Satellite Terminal (MIST)s interconnected to Joint Tactical Training System (JTRS) and Warfighter Information Network-Tactical (WIN_T) networks, as foreseen by Future Combat Systems (FCS). The mentioned battle support vehicles operate as routers and hubs that interconnect Unmanned Air Vehicles (UAV), Unmanned Ground Vehicles (UGV), Apache Helicopters (Ah64) and Land Warriors (LW) with Continental United States (CONUS) based on a wireless C4ISR network infrastructure. The entire operation is directed and controlled via a CONUS based ground station and its corresponding satellite network.

Within this environment, three areas of ES bandwidth and latency research are addressed: *Simulation Traffic Analysis*, *Data Transmission Optimizations*, and *Traffic Modeling Tools / Demonstration sets*. Simulation Traffic Analysis tasks include the development of a tentative network for FCS and Simulator Training systems that can be used to analyze Packet Data Unit (PDU) transmissions of the most critical entity actions and assessment of the *operational-distribution of PDUs*. Future Data Transmission Optimization tasks include the development of *burst-free transmission scheduling*, *PDU replication*, *data compression*, and *OPFOR control hand-off* techniques. Traffic Modeling Tool activities include the creation of *libraries for network capacity planning* and a self-contained *traffic modeling demonstration package* using Omnet++. Within this environment, we present results for *capacity estimates* for ES bandwidth in FCS battle applications.

FCS Bandwidth Optimization Problem.

Over the past decade, the U.S. Army's principal modernization initiative has been its digitization effort, designed to significantly improve the fighting capabilities of soldiers on the battlefield. But implementing that initiative presents significant challenges. Digitization requires the rapid transmission of large amounts of information over significant distances. Experiments conducted to date as well as recent operations in Iraq, where troops employed some of the results of the service's digitization efforts, have shown that that requirement is difficult to fulfill in any terrain conditions.

Consequently, the focus of the Army's modernization program has shifted in 1999 to what it terms transformation—making its forces deployable more quickly while maintaining or improving their lethality and survivability. Although digitization is no longer the Army's primary modernization initiative, it remains a key element of transformation. In the past several years, questions about the size of the information flow associated with digitization and the communications bandwidth to support it, have spurred the Army to adopt several large radio and network communications programs to study the total network capacity of Training Simulations and Real-Time battle communications to predict future FCS design considerations.

Future bandwidth demand shall increase as suggested by Rehmus [1] on his report to the Congressional Budget Office (CBO). He predicts that the peak network demands for the year 2003 are greater at the Brigade and Battalion levels by a factor of 10 to 20 when compared to standard network demands for networks that serve the Operations Officers (ops nets). That is, one message arrives on time for every 10 to 20 sent. Future advances in communications equipment that the Army plans to support include Joint Tactical Radio System (JTRS), Warfighter Information Network-Tactical (WIN-T) and Multiband Integrated Satellite Terminal (MIST) to further support communications at the brigade division and corps command levels increasing further the bandwidth needs. FCS shall exceed the current demands by 10 fold at the Corps and Division Command areas, due to the increase in video and imaging information [5]. In addition, lower communication noticed at other command levels will also increase in the future due to the added support systems and unmanned vehicles planned for FCS use.

Foreseeing the immense bandwidth needs, the Army is trying to reduce its current bandwidth demands by slashing functionality. Broadcasting UAV images, teleconferencing and other bandwidth intensive applications is no longer possible. Useful information has been replaced or eliminated to accommodate the existing network technology such as JTRS and WIN-T. Ironically, decreasing bandwidth needs reduces the success of the Army's digitization Initiative.

The Army faces a number of problems in implementing its IT strategy on the battlefield. The service needs much more bandwidth than it has available today to support both its current systems and those planned for the future. Being Bandwidth the central issue for the communications system, we propose to study the future network requirements. Unfortunately, real time bandwidth measurements are rather complex, particularly when the network topologies are not well defined. To analyze the communication needs we propose to obtain Semi-Automated Forces (SAF) data from the OneSAF Testbed Baseline Simulator (OTB), used by the Army to plan, execute and review battles in remote locations. OneSAF can provide useful data to further study the future network requirements of FCS. Then, using a network constructive discrete simulator such as OmNet++ [2], it is possible to further study the future bandwidth needs and suggest possible optimizations.

Bandwidth considerations for FCS Simulation model.

FCS networks, vehicles and system functionality depend on existing and emergent technologies. Thus, effective bandwidth measurements for future combat systems are difficult due to the inventiveness of the designs. However, certain Bandwidth expectations for certain vehicles are estimated based in information provided by Army Subject Matter Experts (SME) [10]. Data rates have been assigned for certain vehicles for voice, data and imagery. Table 1 lists the effective data rates for FCS vehicles.

FCS Vehicles and Effective Bandwidth.

FCS Vehicles and Support Systems	V:VOICE D:DATA I:IMAGE V:VIDEO	VOICE DATA RATE (Kbps)	VOICE DATA PACKET SIZE (KB)	DATA DATA RATE (Kbps)	DATA PACKET SIZE (KB)	IMAGERY DATA RATE (Kbps)	IMAGERY PACKET SIZE (KB)	VIDEO DATA RATE (Kbps)	VIDEO DURATION (sec)
Aerial Common Sensor	D,I			60	60	60	6144	1000	10
AQF/Prophet	D			30	60			1000	10
AWACS	V	100	20						
Comanche	I			5		5	6144	1000	10
Global Hawk/Predator UAV	D,I			300	60	300	6144	1000	10
J-STARS	NA			NA	NA	NA	NA	NA	NA
Rivet Joint	D			300	60				
Satellite	D,I			600	60	600	6144		
U-2 ASARS II	D,I,V			300	60	300	6144	1000	10
Unmanned Aerial Vehicle (UAV CL I)	D,I,V			0.017	5	10	410	1000	10
Unmanned Aerial Vehicle (UAV CL II)	D,I,V			0.017	5	10	410	1000	10
Unmanned Aerial Vehicle (UAV CL III)	D,I,V			0.017	5	10	820	1000	10
Unmanned Aerial Vehicle (UAV CL IVa)	D,I,V			0.017	5	10	820	1000	10
Unmanned Aerial Vehicle (UAV CL IVb)									
Unmanned ARV-A(L) (2.5 ton)	D,I,V			0.017	5	10	410	1000	10
Unmanned ARV-RSTA (6 ton)	D,I,V			0.017	5	10	820	1000	10
Unmanned Ground Vehicle (MULE)	D,V			0.017	5			192	86400
Unmanned Ground Vehicle (SUGV)	D,I,V			0.017	5	10	820	1000	10

Table 1: FCS Vehicles Effective Bandwidth.

Tanenbaum [7] defines Bandwidth as the range of frequencies transmitted without being strongly attenuated. It can be attenuated as transmission distances increase. Bandwidth units for digital media is known as Bit Rate, the number of bits per second transmitted; not to be confused with Baud Rate the number of signal changes per second. Bit Rate and Baud Rate are related by the following equation.

$$\text{Bit Rate} = \log_2 M * \text{Baud Rate [8]}$$

Therefore, Bandwidth decreases with distance and terrain interference and transmission medium used, an additional channel characteristic that needs to be modeled when building C4ISR network channels. Note that Throughput is analogous to Bandwidth.

Communications traffic can be thought of either approximately continuous or episodic. In the former case, called continuous-flow information (throughput), a bit per second (bps) is the relevant measure; in the later case, referred to as episodic, the size of the message file (in bits) is the appropriate gauge. Table 1 one depicts voice, data, video and imaging throughput for the most common vehicles. Notice that some vehicles transmit voice data only, as the Airborne Warning and Control System (AWACS), while others transmit voice, video and images using the same channel, e.g. UAV.

Building a network simulation using OmNet++ modules to represent FCS network communications is possible. The resultant Bandwidth capacity of the C4ISR based FCS network can be simulated by encoding the corresponding wireless channels and their bandwidth capacity. Satellites, Vehicles and Land Warriors can be modeled as network components with specific data generation characteristics and effective bandwidth. Since all modules in the system transmit in broadcast mode (DIS specification), the overall network throughput and the channel collisions can be analyzed to optimize the available bandwidth. Moreover, channel bottlenecks and slack time can be studied to further optimize the overall throughput. However, simulation and modeling and the software that makes then function is designed according to certain assumptions about the communications network in which they operate and the rates of information available as parameters. Therefore, the results of this experimental simulation are an attempt to provide measurable results and determine the possible network tribulations that future combat systems may present as they intercommunicate through different networks and satellite links in benign environments. Methods to optimize bandwidth utilization such as *burst-free transmission scheduling*, *PDU replication*, *data compression*, and *OPFOR control hand-off* techniques may be used to improve data transmission speeds as results demonstrate network inefficiencies.

Omnet++ Modeling

OMNeT++ is a discrete event simulation environment. Its primary application area is the simulation of communication networks, but because of its generic and flexible architecture, is successfully used in other areas like the simulation of complex IT systems, queuing networks or hardware architectures as well. The simulator provides component architecture for models. Components *(modules)* are programmed in C++, then assembled into larger components and models using a high-level language *(NED)*. Models are provided free of charge [3].

For this particular simulation we choose to model a communications topology based on a battle scenario suggested by the Army SME. The *used case* involves land-unmanned vehicles, air and land support and UAVs, all communicating at a Brigade level. --A brigade is the smallest Army force structure that utilizes a satellite link [1]. A brigade is typically commands the tactical operations of two to five organic or attached combat battalions. Normally commanded by a colonel with a command sergeant major as senior NCO, brigades are employed on independent or semi-independent operations. Armored, cavalry, ranger and Special Forces units are categorized as regiments or groups [2] –.

Four communication channels are necessary and modeled according to the characteristics suggested in C4ISR document for wireless communication [4] and the bandwidth predictions for JTRS (200 Kbps) and WIN_T (2.5 Mbps) networks obtained from [1].

The following figure 1 is generated by the OMNet++ simulator and depicts the current network layout. The first channel is the *wireless ground to satellite (wirelessGS.)* This channel connects CONUS networks with the satellite network that transmits battle command information to remote locations all over the world. The second channel, *wireless to ground network (wirelesWSGN)* supports apache helicopters (AH-64) and Distributed Common Ground Systems (DCGS) vehicles that serve as a router to *WIN_T* networks. The third channel, *WIN_T* connects DCGS vehicles with Manned Platform Vehicles as they also serve as a router for JTRS networks. The last and fourth channel connects wirelessly all Unmanned Ground Vehicles (UGV), Unmanned Air Vehicles (UAV) and Land Warriors (LW) together.

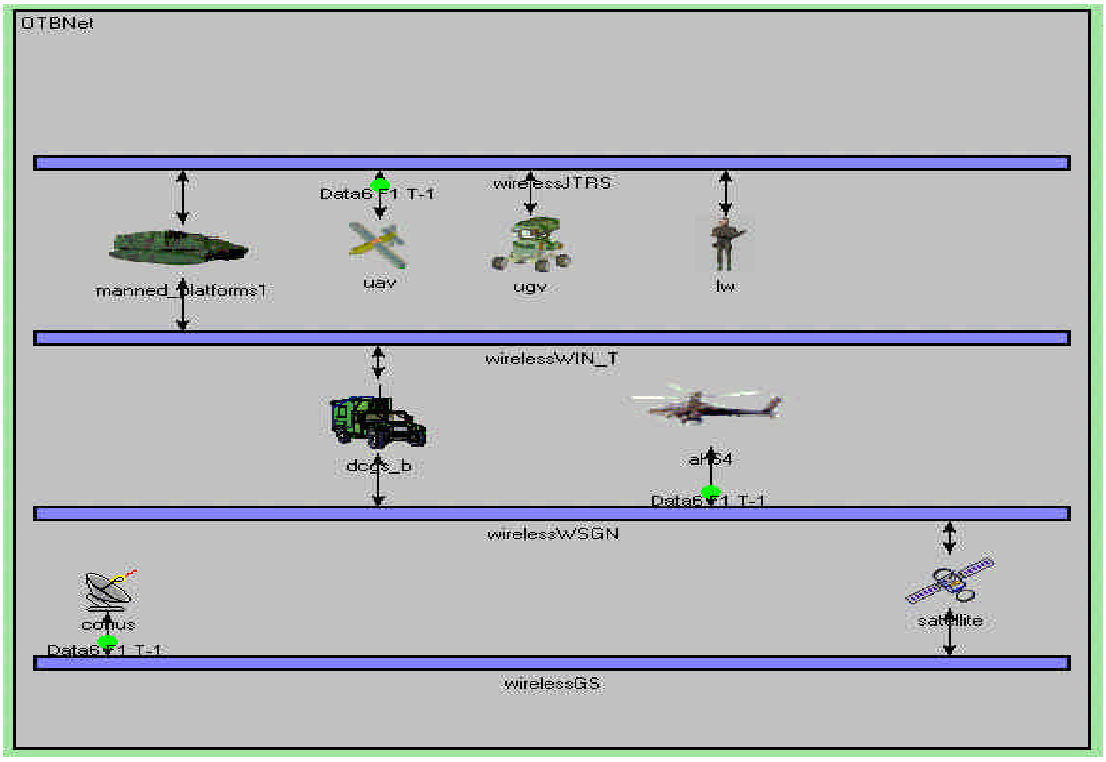


Figure1: OMNet++ C4ISR network channel connections for WIN-T and JTRS networks.

Each channel depicted in blue (elongated rectangles), serve as a bus that transports data from one network channel to the other. Channels are modeled according to the channel characteristics of the protocol. e.g., Wireless LANs use IEEE 802.11 protocol.

Models connect to each channel using *nodes* a sub-module provided by the channel. There is a one to one correspondence between modules and channel nodes. Figure 1 depicts several green colored circles, these are the packets sent by each host generator. Each module is defined according to the desired module specification and characteristics.

A simple module contains a *Generator* and a *Sink*. *Generators* are sub-modules programmed to generate packets at their discrete time only limited by the throughput of the channel it connects to. Generators will broadcast a packet when the packet's time is due. If the packet is to be sent at time t, but the bus due to its limited bandwidth cannot service it, a negative time slack is created and recorded. If the packet leaves on time, a *collision* is detected and recorded. The *Sink* on the other hand, will retrieve packets from the channel with a destination address of –1 (Broadcast Destination) or its own destination (network dependent number). Figure 2, depicts the internal configuration a UAV as shown by OMNett++.

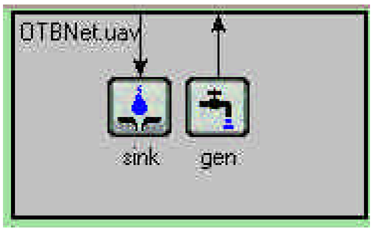


Figure 2: UAV internal sub-modules.

The *Generator* can be programmed to create data packets at a specific data rate and size or it can read data from data text files a rate determined by each in packet's timestamp. When data from an Army simulator is provided such as OneSAF, data can be parsed and reorganized to be read later by the *Generators*. Figure 3 depicts the current data format for packet generation using a text file, therefore for this method; data from OneSAF needs to be parsed accordingly to meet the following requirements.

Column 1 contains packet time information in Hexadecimal 1/100 of a second. The next column contains the packet size information. Original data is converted to generate columns three and four. The *Generator* module reads the data text file and generates Column 3 which contains the converted time in Min:Seconds.Hundreths of a second. Column 4 contains the line number.

positive slack is recorded. If the packet is serviced, but on his way out to the channel collides with an incoming packet, a

0x4f690c7a	32	:18:36.707	1
0x4f7ab058	32	:18:37.676	2
0x4f8ca818	32	:18:38.663	3
0x4ffc8a88	92	:18:44.809	4
0x513da63a	100	:19:02.448	5
0x51752798	92	:19:05.497	6
0x531629f4	92	:19:28.404	7
0x53617074	100	:19:32.539	8
0x548db8ba	92	:19:49.034	9

Figure 3: Data format for packet generation using a text file.

In cases where a single module will generate three different types of data, three *Generators* will be contained in each module, one for each data type and rate.

Once all network components are in place, different network configurations can be explored by rearranging the connections to the channels. Statistical results based on different simulations can be used to aid future designs. The goal is to determine if the current bandwidth utilization is wide enough to accommodate FCS.

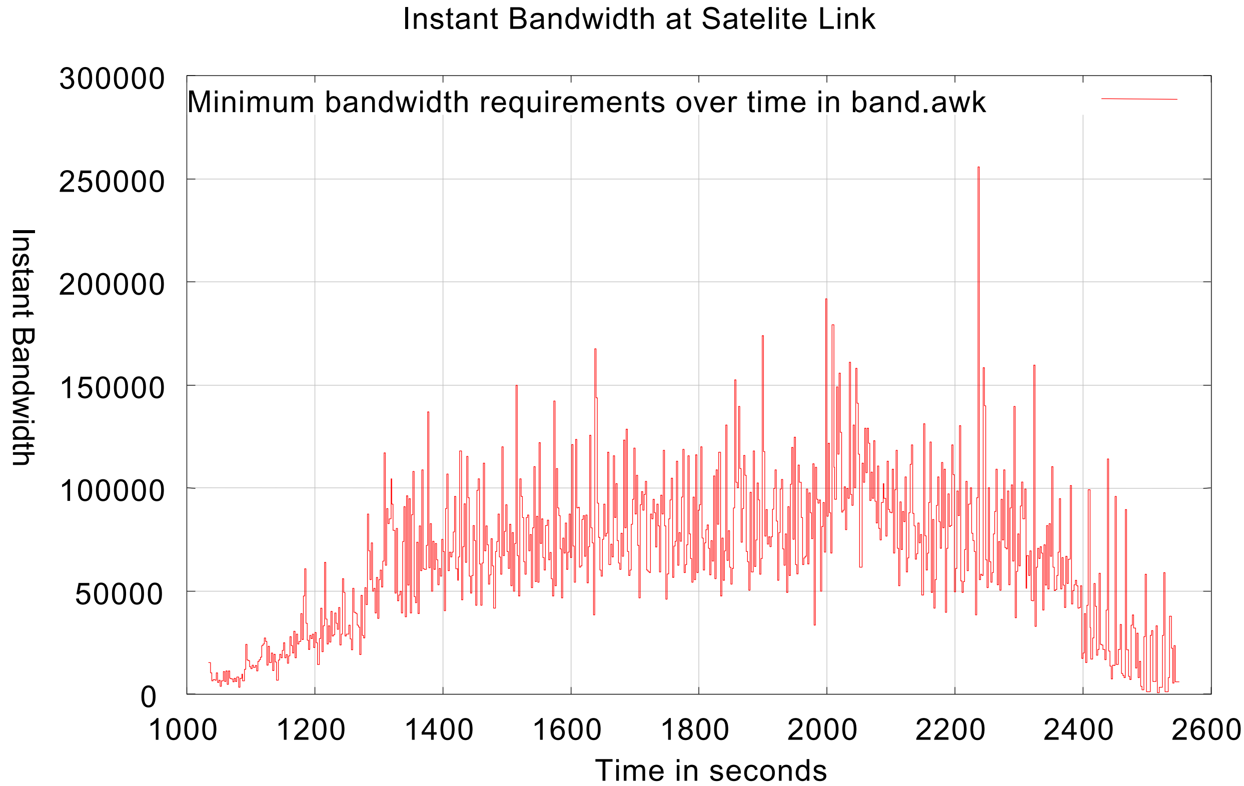
Simulation Results

Peak effective bandwidth demand for future combat systems can exceed the current expectations. The Army has studied current the peak demands for continuous flow-information on division, brigade and battalion levels for the digitized division. The study has found that peak effective bandwidth can be between 2.5 Mbps and 4 Mbps. Our research intends is to find the possible bottlenecks in the system and further optimize the transmission of packets to obtain better bandwidth optimization.

Vehicles such as the Manned Platforms, Army Battle Command System (ABCS) and the DGCS act as centers of communication in the battlefield. Such vehicles act as routers for the JTRS and WIN_T networks and Satellites used in battle at the Brigade, Division and Corps levels. Lower command levels have no use of satellite links, but the battlefield communication is similar. These vehicles are suspects of intense collisions due to the intense routing they perform. The present simulation shall provide collision information on these vehicles as results are obtained from OTB sample data. Unfortunately, the used cases utilized for FCS using OTB have not yet been released as unclassified. Such data and the results of the proposed OMNet++ simulation shall be available prior to the oral presentation.

However, the following graph presented on figure 4, depicts the bandwidth utilization results of an earlier simulation at the satellite module using similar OMNet++ models that supported Joint Tactical Training Systems (JTRS). The OTB vignette supported six C-130 Hercules air carriers on flight and the communication between them as a battle training simulation was executed inside the three vehicles that each plane transports. Data used on this Omnet++ simulation was also obtained from OTB. It is easy to observe that a 200 Kbps channel is necessary at the satellite link to provide optimal service.

Simulation results and an updated paper shall be presented the day of the oral presentation. The suggested data rates depicted on table 1 for the unmanned vehicles are ready to be used and incorporated into the respective modules and provide additional data to the JTRS and WIN_T networks. As we receive the Army OTB unclassified data that represents the bandwidth utilization of our C4ISR proposed network, our simulation shall produce similar results as proved useful in earlier simulations.



References

[1] Rehmus, P. Gilmore, M. (2003) **The Army's Bandwidth Bottleneck** [Electronic Version] *The Congress of the United States, Congressional Budget Office, Washington DC. Retrieved November 19, 2003, from* http://www.iwar.org.uk/rma/resources/cbo/08-28-Report.pdf

[2] **The Army's Force Structure.** *Retrieved November 28, 2003, from* http://www.militarydial.com/army-force-structure.htm

[3] **OMNet++ Discrete Event Simulator System.** *Retrieved on November 26, 2003, from* http://www.omnetpp.org/.

[4] Web, Richard. **C4ISR Architecture Framework 2.0**. *(1997, December) Retrieved on November 26, 2003 from* http://www.defenselink.mil/nii/org/cio/i3/AWG_Digital_Library/pdfdocs/fw.pdf

[5] Francis, Paul L. **Understanding FCS.** (2003) *Retrieved on October 26, 2003 from* http://www.defenselink.mil/nii/org/cio/i3/AWG_Digital_Library/pdfdocs/fw.pdf

[6] Varga, András. **Using the OMNeT++ discrete event simulation system in-ducation.** (1999) IEEE *Transactions on Education, 42(4):372, November 1999. on CD-ROM issue; journal contains abstract.*

[7] Tanenbaum, Andrew S**. Computer Networks.** 4th ed. (2002) *Prentice Hall*

[8] Spragins, John. **Telecommunications: Protocols and Design.** (1992) *Addison-Wesley Publishing Company, Inc.*

[9] André Maurits, George van Montfort, and Gerard vande Weerd. **OMNeT++ extensions and examples.** *Technical report, Technical University of Budapest, Dept. of Telecommunications, 1995.*

[10] Wessel, James - **Subject Matter Expert, MITRE Corporation** (2003) *MITRE-Washington 7515 Colshire Drive, McLean, VA 22102* jwessel@mitre.org

Using Simulation, Modeling and Visualization to Prepare First Responders for Homeland Defense

Magnus Morin[1], Johan Jenvald[1] and Mona J. Crissey[2]

[1] Visuell Systemteknik i Linköping AB,
Storskiftesgatan 21,
SE-583 34 Linköping, Sweden
magnus@vsl.se, johan@vsl.se

[2] U.S. Army PEO STRI
12350 Research Parkway, Orlando, FL, 32826-3276
Mona_Crissey@peostri.army.mil

Abstract

Terrorism with weapons of mass destruction presents numerous challenges to authorities and responders in the public safety area. Training is extremely important to establish and tune an effective and efficient response system capable of handling chemical, biological, radiological and nuclear events. The paper characterizes the unique conditions for training responders from multiple agencies within separate jurisdictions. Examples from four full-scale emergency response exercises cover weapons of mass-destruction, urban environments, and political violence. In each such live simulation exercise, computer support based on modeling and visualization supported feedback and evaluation to promote effective cross-organizational learning.

Introduction

The way we look at public safety has changed in the light of recent world events. Asymmetric warfare is a reality that not only concerns military forces in war zones, but also has the potential to affect the everyday life of millions of people. New types of carriers can deliver weapons of mass destruction in non-traditional ways to inflict casualties and create havoc in civilian communities. Responding to this challenge requires new approaches to crises management and preparedness. Interoperability and integration of services are fundamental to leverage the resources available across both military and civilian organizations in multiple jurisdictions.

In the United States, the federal government has established the Department of Homeland Security. The Department of Homeland Security has the mission to develop a comprehensive national strategy to secure the United States from terrorist threats and attacks and to coordinate its implementation. This strategy identifies emergency preparedness and response as one of six critical mission areas (Office of Homeland Security, 2002). A cornerstone in this area is training. A national training and evaluation system, including a national exercise program, will be established to ensure that civilian personnel at all levels of government are prepared. Training challenges include response to chemical, biological, radiological and nuclear events, military support to civil authorities, multi-agency cooperation, and integrated command and control.

In this paper, we analyze the specific problems involved in training first responders for large-scale operations in response to events involving weapons of mass destruction. Our emphasis is on live simulation exercises. We examine how methods and tools developed around simulation, modeling, and visualization can facilitate this type of training. In particular, we consider how the taskforce training approach by Jenvald (1999) and methods for multimedia representation of tactical operations by Morin (2002) combine to address crucial training issues. A series of examples from previous exercises provides an illustration of difficulties and possibilities identified. Finally, we discuss some future directions for research and development.

Learning from experience

When people train in a live simulation, they take part in experience-based learning. However, participants in complex, dynamic situations are thrown into action with limited possibility to step back and reflect on actions as the situation unfolds (Winograd & Flores, 1986). After the action, on the other hand, it is essential that they reflect on the exercise as a basis for sustaining strengths and remedying weaknesses. Kolb (1984) emphasized the combination of concrete here-and-now experience with the use of feedback to change practices and theories. Norman (1993) noted that reflection on performance makes it possible to better know what to change and what to keep. Effective processing requires accurate feedback on the actions taken, which is often a problem in dynamic and distributed environments, such as rescue operations, where the people may not see the effects of their actions (Hoffman, Crandall & Shadbolt, 1998) and where the environment may change state spontaneously, without deliberate intervention (Wærn, 1998).

Debriefing provides an opportunity to engage in structured reflection on an experience in order to modify behavior based on that experience (Pearson & Smith, 1986; Raths, 1987; Lederman, 1992). In training, debriefing is commonly referred to as after-action review (Morrison & Meliza, 1999). Rankin and associates (Rankin, Gentner & Crissey, 1995) described an after-action review as a professional discussion of an exercise, which concentrates on performance standards. To provide effective feedback, methods and tools to present representations of operations have been developed and used to support after-action reviews in military settings as well as in emergency management and response (Jenvald, 1999). Morin and colleagues (Morin, Jenvald & Thorstensson, 2000) described how models of rescue operations built from multiple sources of data could support analysis and feedback. Applications of this method include training (Crissey, Morin & Jenvald, 2001) and real operations (Morin, 2002; Thorstensson, 2002). It has also been used to investigate communication in command and control (Thorstensson, Axelsson, Morin & Jenvald, 2001; Albinsson & Morin, 2002; Albinsson, Morin & Fransson, 2003).

Training first responders

In the military domain, many simulation-based tools have been developed to enhance the realism of training and to provide effective feedback to the participants (Morrison & Meliza, 1999, Jenvald, 1999). Thorstensson and associates (Thorstensson, Morin & Jenvald, 1999) analyzed the differences between military training and training for emergency management and response and found significant differences, but also many similarities. Large-scale emergency response operations resemble military operations in many respects. Both involve numerous individuals and teams working together in a geographically distributed area of operations. The outcome of an operation depends on the cooperation between individuals and teams and the careful coordination of their efforts. As a consequence, the training issues involved in large-scale emergency response operations are similar to those encountered in the military domain. It is, therefore, important to investigate to what extent existing methods and systems for military training are applicable to the training of first responders in public safety. Table 1 lists some of the differences between the military and public safety domain.

Table 1: Characterization of training conditions in the military domain and the emergency response domain, respectively.

Training issues	Domain	
	Military	Emergency
Structure of the organisation	Homogenous: common culture and background	Heterogeneous: different professions
Main purpose of the training	Improve the ability to solve common tasks	Improve the ability to handle rare situations
Training philosophy	Designated training organisation	No particular training organisation; Main focus on execution of emergency response
Training execution	Sequence of exercises	Single exercise

Morin and colleagues (Morin, Jenvald & Crissey, 2000) analyzed training needs and training opportunities for emergency response to mass-casualty incidents. They identified training audiences and classified training on the individual, team, command post, and taskforce level. In this framework, taskforce training corresponds to full-scale, live exercises with people and equipment in the field. This type of training enables all personnel to engage in a common exercise, where they can apply their skills in a realistic scenario. Such training can produce learning situations at the level of complexity that do not occur in individual training or team training. However, effective taskforce training requires that individuals and teams have attained proficiency in their roles.

Public safety agencies spend most of their time on watch or responding to calls. The time available for training is limited. Furthermore, the public safety community has primarily been organized to handle the everyday accidents that constitute the vast majority of emergencies. Major incidents occur infrequently. Also, the community is heterogeneous in the sense that it includes many different agencies at various levels of government. Major exercises thus must incorporate public safety assets both vertically across levels of government and horizontally across multiple professions and specialties. They are complex, as they require cooperative planning and orchestration.

A primary challenge in training responders for homeland defense is to overcome organizational and jurisdictional barriers. Flin (1996) observed that there seemed to be limited cross-transfer between public safety organizations despite their having similar goals and employing similar means to achieve those goals. Training together and debriefing together may be one way to promote the interaction between agencies.

Training is crucial for preparing first responders for homeland defense. Taskforce training involves teams from multiple agencies and levels of government in a realistic learning situation. To create a realistic training environment and to provide effective feedback, methods and tools originally developed for military training can be adapted to the needs of the civil sector.

Computer-supported taskforce training

To conduct a major exercise that includes an effective after-action review requires a systematic approach to training. Jenvald (1999) presented a method for computer-supported taskforce training, which incorporates steps to ensure that the course of events of the exercise is documented and accessible for after-action review and analysis. Figure 1 gives an overview of the method, which includes the following five steps:

- *Planning*. The planning step defines the goals of the exercise, identifies particular topics or themes that should be highlighted, and creates the exercise scenario accordingly. Evaluation criteria are established relative to the goals and an instrumentation plan is developed. This plan links the goals of the exercise to evaluation criteria by defining observable measures of performance and prescribing means of data collection.
- *Preparation*. Before the exercise, it is necessary to inform participants about the exercise. Instructors and observers need guidelines and instructions regarding what to pay attention to and how to report observations. Automatic instruments for data collection must be installed, configured, and started.
- *Exercise*. During the exercise, the participants act in their roles in the unfolding scenario. Data are captured by the instrumentation system to document the course of events. The result of data collection is a mission history, which is an event-based, time-ordered multimedia model of the mission.
- *After-action review*. Participants, observers, and trainers assemble to conduct an after-action review of the exercise. In a process of critical reflection, the participants explore the exercise using the mission history as a cognitive aid. Instructors and observers may pose questions or offer comments and guidance to facilitate debriefing.

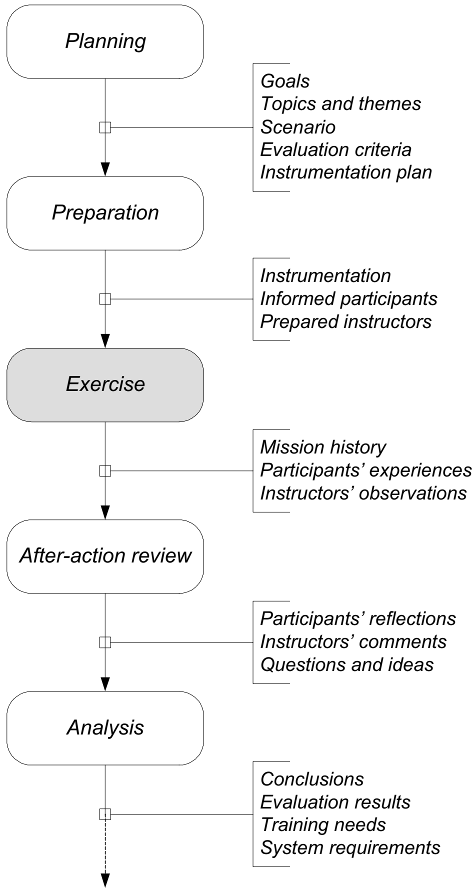


Figure 1: A method for computer-supported taskforce training.

- *Analysis*. Reflections, comments, and remaining questions may serve as starting points for a deeper analysis of the exercise. Evaluation results and observations may lead to new training needs and modified tactics, organization, and equipment.

Both the after-action review and the post-mission analysis are exploratory tasks. Observations during the exercise and findings from examining the mission history lead to new questions that can be explored. To support this mode of analysis, the MIND presentation tool includes an exploratory multimedia interface. This interface enables a user to browse the mission history and to replay sequences of the mission (Morin, 2002).

Application examples

We describe four examples of how computer-supported taskforce training applied to emergency management and response in scenarios addressing weapons of mass-destruction in multi-agency settings. The examples come from exercises: *Alvesta*, *Orlando*, *Cornelia*, and *Daniela*.

Alvesta. The first field exercise took place in Alvesta, Sweden, in October 1997 as a part of a project sponsored by the Swedish Rescue Services Agency. The goal of this project was to develop methods and tools for assessing emergency planning for incidents involving chemical warfare agents (see Figure 2). An important activity in this project was to conduct operations based on realistic scenarios according to existing plans and study the outcome (Rejnus, Jenvald & Morin, 1998). The exercise involved emergency response to a simulated chemical attack on the town Alvesta. Approximately 180 first responders from different agencies participated and 49 extras acted as casualties. For this exercise, methods and tools were transferred from the military domain to civilian emergency response (Thorstensson *et al*., 1999). Data collected included:

- Position tracks from GPS locators
- Deployment of special resources (decontamination station and medical aid station)
- Deployment of units in clean and contaminated areas
- Reports from chemical detection
- Casualty flow and treatment data using timed checkpoints
- Tactical radio communication
- Digital photographs
- Video footage

The mission history from Alvesta was used to explore the exercise at an after-action review for all 230 participants 90 minutes after it ended. The exercise commander from the Swedish Rescue Services Agency acted as a tour guide and conducted a walkthrough of the exercise with frequent stops at critical situations. Key participants commented on situation assessment and decisions, using the mission history both to identify and recall situations and to illustrate their line of thought. Two weeks later the model was used in a similar manner to support an in-depth analysis of the different functions in the operation as a basis for assessing the emergency plans. Finally, the mission history was included in a multimedia CD-ROM addressing chemical warfare hazards and used in a curriculum covering nuclear, biological, and chemical hazards at Umeå University, Sweden (Jenvald, Morin & Rejnus, 2000).

Orlando. In May 2000 a joint Swedish-American team organized a field study in Orlando, Florida (Crissey, Morin, & Jenvald, 2001). The purpose was to explore the effects of transferring the methods and tools used in the Alvesta exercise to a different country, organization, and culture. To this end the team participated in an emergency-response exercise to a chemical incident. In the operation fire–rescue units from two counties and

Figure 2: Swedish rescue personnel decontaminate a victim of the nerve agent VX (Photograph by Johan Jenvald).

Figure 3: Orange County firefighters in level-A suits work to contain a chlorine leak (Photograph by Bo Tingland).

three police forces, drawn from a 25- by 25-kilometer area, responded to a simulated chlorine gas leak in a suburban location (see Figure 3). There, they rescued and treated two victims and contained the leak. The operation took place in the morning and after a lunch-break the participants assembled for an after-action review. The Director of Training of the Orange County Fire Rescue Department facilitated a walkthrough of the operation using the mission model along the same lines as in the Alvesta study. The Orlando exercise in many respects was a replication of the Alvesta exercise. The same methods and tools were used. Models and views could be reused once the geodetic differences were settled. Reconstruction and exploration supported an after-action review similar to that in Alvesta. In spite of differences in legislation, doctrine, organization, culture, and language, we found nothing in the study that contradicted the assumption that the methods and tools for reconstruction and exploration could indeed be transferred to different environments.

Cornelia. On November 22, 2000, some 200 first responders and command staff from the local fire department, the county medical services, and county police joined forces with personnel from the urban transport authorities and train operators to practice emergency response to a subterranean train derailment. The scenario centered on a train derailment in the Stockholm subway system, caused by sabotage. In a tunnel, 150 meters from the platform of a downtown subway station, an object on the track caused a train to derail, hit the tunnel wall, and come to an abrupt stop. The impact left 86 people on the scene with various injuries, including five fatalities represented by mannequins. The train driver, who had sustained only slight injuries, used the train's radio to notify the traffic control center about the incident. From that point, personnel in the traffic control center initiated emergency activities according to their standard procedures. All victims were removed from the train, stabilized on scene, and transported to three trauma hospitals in the Stockholm area.

Using similar data collection procedures as in the previous two cases a mission history was constructed from the exercise. The mission history supported an after-action review three hours after the exercise ended. Furthermore, it was crucial in the analysis of coordination and collaboration among responders from different agencies. The starting point of the analysis was questions raised by participants and observers at the after-action review. However, several new issues were discovered in the course of analysis. For example, why did it take so long to get the victims to the hospitals? The estimated time was 90 minutes, but in reality it took 2 hours and 43 minutes. The analysis leading to a tentative answer to this question provides an illustrative example of how the mission history can support exploration of inter-agency coordination. Table 2 summarizes the information used in this process.

- Items 1 and 2 define the scope of the analysis by identifying the start of the exercise and the time of arrival of the *last* patient.
- Items 3 and 4 identify when the *first* patient left the incident scene and arrived at the hospital.

Was there a plausible explanation for the 67-minute period that elapsed before the first patient left the subway station? To answer this question, we examined how the chain of medical aid was organized. A critical strategic decision is whether to establish a casualty collection point (CCP), or to rush patients to hospitals with only minmal on-scene prioritization and stabilization. The medical commander is responsible for this decision, but needs to consult the fire commander regarding a suitable location for a CCP. In either case, the police are responsible for registering the casualties.

Figure 4: The deputy fire commander meets the police officer in charge of the station platform (Photograph by Lars Rejnus).

- A photograph from the ticketing area at 10:14 shows police officers preparing a CCP (Item 5). Browsing police communication around 10:14 revealed Item 6, an audio link that suggests that fire commander was involved in the decision. Items 7 and 8 illustrate the growing confusion regarding the CCP in the police organization. The police commander 1050, at the joint incident command post, and officer 1710, in the ticketing area, clearly had diverging views.
- Going back to the first minutes of the operation revealed that the police were first on scene (Item 9) and started organizing their activities. The first medical units arrived 7 minutes later followed by the first fire–rescue units (Items 10 and 11). The senior fire commander arrived at 10:16 (Item 12).

A shift of command took place in the fire–rescue organization when the senior fire commander arrived around 10:16. The first commander (from unit 123) may have failed to report the crucial decision about the CCP to the senior commander. Alternatively, he may have regarded the CCP in the ticketing area as a temporary assembly point. The latter hypothesis is supported by data from the mission history.

- An audio clip (Item 13) contains an order to deploy tents for a CCP in a park outside the subway station. A video clip from 10:40 shows the incident commander discussing the CCP with his deputy. In this conversation, they mention three CCPs, two of which are regarded as temporary (on the platform and in the ticketing area).

Table 2: Exploration of the ambiguous casualty collection point using the Cornelia mission history.

Item	Time	Medium	Source	Description
1	10:00	Audio	Traffic control network	The train driver reports the incident to the traffic control center
2	12:43	Report card	Casualty registration	The last two victims arrive at Danderyd Hospital.
3	11:07	Photo	Medical observer	Ambulance crew bring the first patient out from the station
4	11:13–11:25	Position track	GPS	Ambulance B887 takes the first patient to the hospital
5	10:14	Photo	Police observer	Police officers prepare a CCP in the ticketing area
6	10:15	Audio	Police network	The officer in charge of casualty registration (1710) reports to the police commander (1050) that the fire commander has ordered the CCP to be in the ticketing area. This decision has been communicated to the medical personnel on scene.
7	10:24	Audio	Police network	1050 informs 1710 that the CCP is going to be in tents in the street outside the station
8	10:30	Audio	Police network	1710 requests from 1050 a clarification of the location of the CCP, because medical personnel are going to remain in the ticketing area
9	10:03	Position track	GPS	Police unit 9770 arrives, followed by police units 9760, 1710, and police commander 1050
10	10:10	Position track	GPS	Emergency medical team B880 and ambulance B881 arrive
11	10:11	Photo	Fire observer	The first fire units arrive. The officer of unit 123 assumes command
12	10:16	Photo	Fire observer	The ranking fire commander (102) arrives and assumes command
13	10:23	Audio	Fire command network	102 orders fire unit 435 to deploy tents for the CCP in the street outside the station
14	10:40	Video	Command post observer	The incident commander discusses the situation with his deputy and summarizes his view of the CCP. Three CCPs are mentioned: platform, ticketing area, and tents outside the station.
15	10:39	Audio	Fire tunnel network	Fire unit 193 asks the fire commander 102 whether the CCP in the street is ready to receive casualties.
16	10:49	Photo	Medical observer	Four ambulances idling outside the station with drivers

- Another audio sequence (Item 14) reveals that the fire commander in charge of the platform is waiting for a CCP in the street. He does not mention the CCP in the ticketing area.

Personnel from the different agencies did not have a common view of how the chain of medical attendance was going to be organized. Item 8 clearly illustrates this confusion from the point of view of the police. Also, Item 8 indicates that the medical personnel had no intention to relocate to tents in the street.

Although sometimes confusing and contradictory, Items 5 to 15 suggest that the operation was going to include a CCP. However, Item 16 indicates that the medical services did not share this view. It shows ambulances idling with drivers ready to pull up to the station entrance to load patients. Unfortunately, there was no evidence in the mission history about who made the decision to keep ambulance crews waiting in the street.

The example is representative for the typical course of analysis in an explorative setting. A piece of data triggered a question. Another piece provided a clue and a time point. Browsing communication data around that time point revealed additional relevant data. Multiple sources made it possible to corroborate findings and construct a chain of evidence to formulate hypotheses and support conclusions. The mission history was a crucial element in this analysis by presenting the basic facts of the tactical operation. MIND facilitated exploration by supporting access and navigation in the mission history. In particular, it helped analysts link communication data to contextual information and vice versa.

Daniela. In December 2002, first responders from various agencies in Linköping, Sweden, trained together in a scenario that involved politically motivated violence. A covert political meeting turned into a blazing fire, when militant members of an opposing fraction appeared on the scene (see Figure 5). This scenario was constructed to force commanders from various agencies to integrate command and control, while the focus of the incident shifted from law enforcement, to fire suppression and rescue, and to medical aid.

During the exercise in the morning, observers and technical systems collected data. In the afternoon, participants and observers conducted an after-action review facilitated by the mission history. Later, the agencies involved used the mission history to analyze incident command, coordination, and communication. This use of computer-supported taskforce training supported the conclusion from the previous exercises: Exploration and context are crucial for reconstructing the events of a complex scenario to facilitate reflection and analysis.

Figure 5: A demonstrator with a smoke bomb defies attempts to restore law and order (Photograph by Johan Jenvald).

Discussion

Although complex and resource intensive, live simulation exercises are essential for developing and sustaining the preparedness and responsiveness of our first responders. They provide an opportunity for managers and responders to gain an insight into the processes involved in a major operation and their interactions. With appropriate performance measures, live simulation can be used to validate the incident response system (Rejnus *et al*., 1998). However, Jenvald (1999) cautioned us not to confuse training and system validation, because they have different ends and means.

Feedback is a critical component in experience-based learning. In a distributed environment, people have problems relating actions to outcomes (Hoffman *et al*., 1998). Computer-supported taskforce training operates under the assumption that providing a coherent view of the course of events is a crucial step toward overcoming this difficulty. Modeling and visualization are key techniques in constructing a mission history and exploring it in an after-action review as well as in subsequent analyses (Morin, 2002). A mission history is a tangible and visible representation of an operation that can be examined and disseminated. As such, it can convey lessons learned from operations to responders who did not participate.

The examples included in this paper indicate how the methods and tools of computer-supported taskforce training apply to training of first responders for homeland defense. Dealing with weapons of mass-destruction in urban environments is a nightmare scenario that must be seriously considered. Such events never entail just a fire–rescue mission or a law enforcement operation. Instead, multiple agencies must join forces within their jurisdictions to handle the situation. The Cornelia example demonstrates some of the difficulties involved in inter-agency coordination. It also gives ideas for future research on how explorative analysis can be conducted using process data and multiple levels of representations (see also Woods, 1993; Xiao & Vicente, 1999).

The need for training is perpetual. Acquiring new skills and new knowledge, sustaining proficiency, and practicing to achieve higher levels of competence must be inherent activities in any response organization. To this end, public safety agencies must analyze training needs and identify training opportunities to devise appropriate training programs (Morin *et al*., 2000). Taskforce training is one crucial element in a training approach for homeland security. In our field exercises, we have observed that the after-action review process is greatly facilitated by the use of a mission history. The trainees were able to relate their own performance to the activities of other teams and individuals. Usually, the trainees accepted the mission history presented as a fair and unbiased description of the course of events. As a result, the discussions at the after-action reviews tended to be very open-minded and focused on facts. We have found that this positive environment promotes rapid learning by facilitating the process of identifying strengths and shortcomings in individual, team and management procedures.

References

Albinsson, P.-A. & Morin, M. (2002). Visual exploration of communication in command and control. In *Proceedings of the Sixth International Conference on Information Visualization* (IV 02), pp. 141–146, 10-12 July, London, England.

Albinsson, P.-A., Morin, M. & Fransson, J. (2003). Finding information needs in military command and control systems using exploratory tools for communication analysis. In *Proceedings of The Human Factors and Ergonomics Society's 47th Annual Meeting*, (HFES 2003), pp. 1918–1922, October 13–17, Denver, Colorado.

Crissey, M. J., Morin, M. & Jenvald, J. (2001). Computer-supported emergency response training: Observations from a field exercise. In *Proceedings of the 12th International Training and Education Conference*, ITEC'2001, pp. 462–476, Lille, France.

Flin, R. (1996). *Sitting in the hot seat*: *Leaders and teams for critical incident management*. Chichester: Wiley.

Hoffman, R. R., Crandall, B. & Shadbolt, N. (1998). Use of the critical decision method to elicit expert knowledge: A case study in the methodology of cognitive task analysis. *Human Factors*, 40(2), 254–276.

Jenvald, J. (1999). *Methods and tools in computer-supported taskforce training*. Linköping Studies in Science and Technology, Dissertation No. 598, Linköping: Linköpings universitet.

Jenvald, J., Morin, M. & Rejnus, J. (2000). Developing Digital Courseware from Multimedia Documentation of Full-Scale Chemical Exercises. In: *Proceedings of The NBC 2000 Symposium on Nuclear, Biological and Chemical Threats in the 21st Century*, June 13-15, Espoo, Finland.

Kolb, D. A. (1984). *Experiential learning: Experience as a source of learning and development*. Englewood Cliffs: Prentice Hall.

Lederman, L. C. (1992). Debriefing: toward a systematic assessment of theory and practice. *Simulation & Gaming*, *23*(2), 145–160.

Morin, M. (2002). *Multimedia representations of distributed tactical operations*. Linköping Studies in Science and Technology, Dissertation No. 771, Linköping: Linköpings universitet.

Morin, M., Jenvald, J. & Crissey, M. J. (2000). Training needs and training opportunities for emergency response to mass-casualty incidents. In: *Proceedings of The 11th International Training and Education Conference* (ITEC'2000), April 11-13, The Hague, The Netherlands.

Morin, M., Jenvald, J. & Thorstensson, M. (2000). Computer-supported visualization of rescue operations. *Safety Science*, *35*(1-3), 3–27.

Morrison, J. E. & Meliza, L. L. (1999). *Foundations of the after action review process*. Special report 42, Alexandria: United States Army Research Institute for the Behavioral and Social Sciences.

Norman, D. A. (1993). *Things that make us smart*. Reading: Addison-Wesley.

Office of Homeland Security (2002). National Strategy for Homeland Security.

Pearson, M. & Smith, D. (1986). Debriefing in experience-based learning. *Simulation/Games for Learning*, *16*(4), 155–172.

Rankin, W. J., Gentner, F. C. & Crissey, M. J. (1995). After action review and debriefing methods: Technique and technology. In *Proceedings of the 17th Interservice / Industry Training Systems and Education Conference,* pp. 252–261, Albuquerque, New Mexico.

Raths, J. (1987). Enhancing understanding through debriefing. *Educational Leadership*, *45*(2), 24–27.

Rejnus, L., Jenvald, J. & Morin, M. (1998). Assessment of emergency planning based on analysis of empirical data. In: *Proceedings of the Sixth International Symposium on Protection against Chemical and Biological Warfare Agents* (CBWPS), pp. 377–383, May 10-15, Stockholm, Sweden.

Thorstensson, M. (2002). Data collection in rescue operations. In R. T. Newkirk (Ed.), *The International Emergency Management Society 9th Annual Conference Proceedings*, pp. 136–147, May 14-17, Waterloo: University of Waterloo.

Thorstensson, M., Axelsson, M., Morin, M. & Jenvald, J. (2001). Monitoring and analysis of command-post communication in rescue operations. *Safety Science*, *39*(1–2), 51–60.

Thorstensson, M., Morin, M. & Jenvald, J. (1999). Extending a battle-training instrumentation system to support emergency response training. In: *Proceedings of the 10th International Training and Education Conference* (ITEC'99), pp. 550–562, The Hague, The Netherlands.

Wærn, Y. (1998). Analysis of a generic dynamic situation. In Y. W Wærn (Ed.) *Co-operative process management*: *Cognition and information technology*, pp. 7–19, London: Taylor & Francis.

Winograd, T. & Flores, F. (1986). *Understanding computers and cognition*: *A new foundation for design*. Norwood: Ablex.

Xiao, Y. & Vicente, K. (1999). A framework for epistemological analysis in empirical (laboratory and field) studies. *Human Factors*, *42*(1), 87–101.

Woods, D. D. (1993). Process-tracing methods for the study of cognition outside of the cognitive psychology laboratory. In G. A. Klein, J. Orasanu, R. Calderwood & C. E. Zsambok (Eds.), Decision making in Action: Models and methods, pp. 228–251, Norwood: Ablex.

Methodological Aspects of M&S

Parameterization as a Method to Simulate the Behavior of Mathematical Models

Dalia M. Gil

Department of Electrical Engineering and Computer Engineering
University of Central Florida
P.O. Box 162450. Room ENG 407, FL 32816, USA
dgil@mail.ucf.edu

Abstract

Parametric analysis has been shown to simulate the behavior of linear mathematical models applied to the solution of many economical problems which decision-making should be conducted under conditions of risk. A certain class of these problems known as transportation type problems arises very frequently in practical applications; two of them are the standard transportation problem and the generalized transportation problem. Once the optimal solution of a mathematical model is obtained, parameterization is useful to know how it will change if some of parameters are changed. One way to attenuate indetermination and to facilitate the process of decision-making is the simulation of perturbations that affect the model applying techniques of parametric programming. The parametric analysis is an important tool to fight the uncertainty. The present paper describes the algorithms that simulate changes for the optimal solution of models with structure analogous to transportation problems (generalized and standard). The method proposed shows the limits of variation to different parameters, which will not affect the optimal solution, and how parameter values outside of these limits will change the current solution. The parameters analyzed are the cost shipping, associated to variables, the quantity of the product available and the quantity of the product required. The method developed has been applied with a few modifications to the solution of different analogous problems to load transportation balancing modeling. These applications have been used as study cases for students of Industrial Engineering. The prototype software was programmed in C language.

Introduction

Mathematical models can be developed to represent real problems. Many of these are called optimization problems because they are focused on maximizing or minimizing a numerical function with a number of variables. When the relations among the variables are linear, the problems to solve belong to the Linear Programming field. A certain class of these problems known as transportation type problems arises very frequently in practical applications; two of them are the standard transportation problem and the generalized transportation problem. Transportation problems have been applied to practical problems in economics, industrial, military and management operations. Once the mathematical model and its solution are obtained, different events can occur, such as changes in costs or requirements. Furthermore, some constraints are omitted or new constraints are incorporated into the model. In all these cases, it is necessary to know how new or perturbed parameters could change the optimal solution obtained. These problems are called post-optimization problems and constitute an important tool to simulate the behavior of the modeled system.

The present paper describes two algorithms and the software that simulate changes for the optimal solution of models with structure analogous to both transportation problems (generalized and standard). The method proposed shows the limits of variation to different parameters, which will not affect the optimal solution, and how parameter values outside of these limits will change the current solution. The parameters analyzed are the cost shipping, associated to variables, the quantity of the product available and the quantity of the product required.

The method developed has been applied with a few modifications to the solution of different analogous problems to load transportation balancing modeling. These applications have been used as study cases for students of Industrial Engineering. The prototype software was programmed in C language.

The Generalized Transportation Problem

The generalized transportation problem can have the following form:
Maximize or minimize the linear function:

$$Z = \sum_{i}\sum_{j} C_{ij} X_{ij}$$

Subject to the following constrains:

$$\sum d_{ij} X_{ij} \pm X_{si} = a_i, \text{ where } a_i > 0, i = 1,m \quad (1)$$
$$\sum X_{ij} = b_j, \quad \text{where } b_j > 0, j = 1,n$$
$$X_{ij} \geq 0, \; X_{si} \geq 0$$

Where

d_{ij} : Input-output coefficients of X_{ij}, they are arbitrary numbers rather than unity.
X_{si} : Slack or surplus variables.
X_{ij}: The quantity shipped from origin i to destination j.
a_i : The quantity of the product available at origin i.
b_j : The quantity of the product required at destination j.
C_{ij}: The cost of shipping of one unit from origin i to destination j.

The table 1 shows the tabular form of the generalized transportation problem. O_i and D_j represent an origin and destination points, respectively. Each O_i indicates that the row i pertains to origin i. Similarly, each D_j as a heading for column j indicates that this column pertains to destination j.

Table 1: Tabular model of the generalized transportation problem

	D_1		D_2			D_n		Slack or Surplus	a_i	u_i
O_1	c_{11} / d_{11}	x_{11}	c_{12} / d_{12}	x_{12}	...	c_{1n} / d_{1n}	x_{1n}	x_{s1}	a_1	u_1
O_2	c_{21} / d_{21}	x_{21}	c_{22} / d_{22}	x_{22}	...	c_{2n} / d_{2n}	x_{2n}	x_{s2}	a_2	u_2
	⋮		⋮		...	⋮		⋮	⋮	
O_m	c_{m1} / d_{m1}	x_{m1}	c_{m2} / d_{m2}	x_{m2}	...	c_{mn} / d_{mn}	x_{mn}	x_{sm}	a_m	u_m
b_j	b_1		b_2		...	b_n				
v_j	v_1		v_2		...	v_n				

Simulation by Means a Parametric Analysis

The simulation by means a parametric analysis starts out with the optimal basic feasible solution in the generalized transportation problem.

$$Z_{ij} - C_{ij} \leq 0, \; \forall \; i, j \quad (2)$$
$$\text{If} \quad X_{ij}^{b} = C^{b}_{ij} B^{-1}$$

Now:

$$Z_{ij} - C_{ij} = C^{b}_{ij} B^{-1} P_{ij} - C_{ij} \leq 0 \quad (3)$$
$$(U, V) = C^{b}_{ij} B^{-1}$$
$$P_{ij} = d_{ij} * e_i + e_{m+j}$$

When solution is found, the corresponding dual problem of the generalized transportation is:

$$d_{ij} * u_i + v_j \leq C^{b}_{ij} \quad i = 1, \ldots, m; \; j = 1, \ldots, n.$$
$$u_i \geq 0, \quad i = 1, m$$

$$\text{Max } Z = a_i * u_i + b_j * v_j$$
$$d_{ij} * u_i + v_j = C^{b}_{ij} \quad (4)$$
$$u_i = 0$$

They are (m+n) equations. An equation of the form $u_i = 0$ appears if the slack or surplus variables X_{si} is in the primal basis. Where
(U, V): Shadow prices.
e_k: Unit vectors for E^{m+n}.
C^{b}_{ij}: Shipping costs associated with the basic variables.
Z_{ij}: Profit.
X_{ij}^{b}: Basic variable.
B^{-1}: Inverse of the basis matrix B.
P_{ij} : Activity vector that corresponds to X_{ij}.

Once the generalized transportation problem has been solved, a post-optimality analysis could be performed, and we can know how the solution will change if some of the parameters, such as prices or the capacities or requirement vectors are changed.

The following problems will be discussed:

- Changing the cost vector.
- Changing the requirement and capacity vectors.

Changing the cost vector

Now let us assume again the optimality condition formulated in (3)

$$Z_{ij} - C_{ij} = C^{b}_{ij} B^{-1} P_{ij} - C_{ij} \leq 0$$

When C_{ij} or C^b_{ij} are changed, the optimality condition can be formulated as:

$$Z_{ij} - C_{ij} = (C^b_{ij} + \Delta C^b_{ij}) B^{-1} * P_{ij} - (C_{ij} - \Delta C_{ij}) <= 0 \quad (5)$$

The aim is to determine the largest values of ΔC^b_{ij} and ΔC_{ij} for which the given optimal basic solution remains optimal.

The present section will analyze two particular cases of changing the costs. These are:

- Changing the cost associated with a basic variable (C^b_{ij}).
- Changing the cost associated with a not basic variable (C_{ij}).

Changing the cost associated with a basic variable (C^b_{ij})

When a change takes place in a C^b_{ij} corresponding to a basic variable, the shadow prices u_i or v_j are varied, since they are function of the following equalities:

$$d_{ij} * u_i + v_j = C^b_{ij} \quad \text{, for all i,j}$$

This system of equations will have variations in one of its variables. The values of u_i and v_j will be calculated again, and $Z_{ij} - C_{ij}$ of each cell associated with a not basic variables X_{ij} will be analyzed to observe whether the solution maintains condition of being optimal. In order to make this analysis, only prices associated with basic variables can change (**ΔC_{ij}** = 0).

Then the optimality condition can be formulated as:

$$\begin{aligned} Zij - C_{ij} &= (C^b_{ij} + \Delta C^b_{ij}) B^{-1} P_{ij} - C_{ij} \leq 0 \\ &= (Zij - C_{ij}) + \Delta C^b_{ij} B^{-1} P_{ij} - C_{ij} \leq 0 \end{aligned}$$

Where

$Y_{ij} = B^{-1} P_{ij}$ Vector image of the P_{ij} that obtains in each iteration.

Then

$$\underset{Y_{ij} > 0}{\text{Max}} - (Z_{ij} - C_{ij})/ Y_{ij} \leq \Delta C^b_{ij} \leq \underset{Y_{ij} < 0}{\text{Min}} - (Z_{ij} - C_{ij})/ Y_{ij}$$

The new value of C^b_{ij} is expressed as:

$$C^b_{ij\,(new)} = C^b_{ij\,(old)} + \Delta C^b_{ij} \quad (7)$$

Where:

$C^b_{ij\,(new)}$ = new value of C^b_{ij} analyzed.

$$C^b_{ij\,(new)} = C^b_{ij\,(old)} + (1 + \Delta C^b_{ij} / C^b_{ij\,(old)})$$

Where:

$$L = \Delta C^b_{ij} / C^b_{ij(old)} \quad \text{(relative increment)}$$

Then

$$C^b_{ij(new)} = C^b_{ij(old)} (1 + L) \quad (6)$$

The interval of study object is

$$0 \leq C^b_{ij(old)} (1 + L) \leq {}_{+}\infty$$

$$-1 \leq L \leq {}_{+}\infty$$

On this basis of previous premises, an efficient algorithm was designed. This algorithm is composed of the following steps:

1. Determine the optimal basic solution of the generalized transportation problem.
2. Select one C^b_{ij} and obtain its new value in the equation (6).
3. Determine the shadow prices from $d_{ij} * u_i + v_j = C^b_{ij(new)}$
4. Determine the values of $Z_{ij} - C_{ij}$ as function of L. As a result, we obtain a set of values L. They can be positive and negative values.
5. Select the minimum for the positive values of L and the maximum for the negative values of L.
6. Change at the optimal solution the L-values chosen and determine the limits of original basic optimal solution.

Changing the cost associated with a not basic variable (C_{ij})

In order to make this analysis, only prices associated with not basic variables can change ($\Delta C^B_{ij} = 0$).

$$\begin{aligned} &Z_{ij} - C_{ij} \leq 0 \\ &Z_{ij} - C_{ij} = d_{ij} u_i + v_j - C_{ij} \\ &Z_{ij} - C_{ij} = d_{ij} u_i + v_j - (C_{ij} - \Delta C_{ij}) \leq 0 \\ &\Delta C_{ij} \geq d_{ij} u_i + v_j - C_{ij} \end{aligned}$$

When the value of C_{ij} associated with a not basic variable changes, values of dual variables don't change because the corresponding system of equations has no variation in its members. In this case the algorithm has the following steps:

1. Determine the optimal basic solution of the generalized transportation problem.
2. Select one C_{ij} to analyze, and calculate the interval of variation.

As the reader can observe, in this case only the decrement of C_{ij} is interesting because when the value of this cost increases, the optimal basic solution does not change.

Changing the Requirement and Capacity Vectors

Frequently, using the dual simplex method solves the parametric analysis of requirements b_j or capacities a_i vectors in linear programming problems. In the present paper a more efficient method based on a perturbation is proposed. The method consists in establishing of an arbitrary infinitesimal amount called L. It is established with the objective of discovering those of the generalized transportation problem cells that are affected when his value is constantly changing. In the present paper only is explained the changes associated with the requirement vector because the analysis is similar to the capacity vector.

Assume that we have found the optimal solution of the generalized transportation problem, and that b_j will be changed by means of the amount L

$$L = \Delta b_j / b_{j(old)}$$

Where

$$b_{j(max)} = b_{j(old)} (1 + L)$$

The limit values for L are determined by means of the following expressions:

$$\text{Max} - X^b_{ij(old)} / B_{ij} \quad \leq \Delta b_j \leq \quad \text{Min} - X^b_{ij(old)} / B_{ij}$$

Where:

B_{ij} are the coefficients of the basis matrix.
$X^b_{ij(old)}$ is the element i of the old basis solution X^b.

In the present work, given the particular characteristics of the generalized transportation problem, a simple parametric analysis method is proposed. Method proposed taking in consideration the existence of open capacities, so slack variables with value different from zero are in the optimal solution. This consideration is very important because the slack variables are precisely which give the potential to increase demands. In this case, we only are analyzed the increment because the idea is to plan how the requirements are satisfied.

The algorithm is:

1. Determine the optimal basic solution of the generalized transportation problem.
2. Select one b_j to analyze and find the cells associated with this column where basic variables are, then select which of these variables are on rows with slack variables different from zero.
3. If there are more than one cells with the characteristics described above, select which has lower cost.
4. Choose the perturbed value for L
5. Calculate the critical value of L that reduces to zero the corresponding slack on the selected row.

$$bj_{(max)} = b_{j\ (old)}\ (\ i + L)$$

6. Don't perturb the value of b_j if there isn't a slack variable associated to the cell with a basic variable.

The parametric method was extended to standard transportation problem where "a product is available in known quantities at each of m origins. It is required that given quantities of the product be shipped to each of n destination. The minimum cost of shipping a unit of the product from any origin to any destination is known. We wish oriented to determine the shipping schedule, which minimizes the total cost of shipment" [Hadley, 1963].

One characteristic of the standard transportation problem is that the coefficients of the inverse matrix take values ±1, 0, and the division does not affect the calculation. As a result, the parametric method developed above can be applied to the standard transportation problem with a little change. That is the coefficients d_{ij} take value 1.

Both algorithms were applied with a few modifications to the solution of different analogous problems: the design of the computer-shared system at Havana Polytechnic Institute [Garcia, 1988], Load Transportation Balancing Modeling for Havana Port operations [Garcia et al, 1991], and Load Transportation Balancing Modeling for the Cuban Ministry of Transportation [Garcia et al, 1993]. Furthermore, these applications have been used for many years as study cases for students of Industrial Engineering and Computer Engineering at Havana Polytechnic Institute. The software was programmed in C language

To illustrate the method proposed, we shall briefly define a general case study on which the author worked in cooperation with company personnel. No actual data will be presented, and the problem will be reduced.

A general Case Study

There is a set of cargo vessels that should be loaded with different merchandises to be shipped. There is also excess capacity in each ship. The total available capacity is greater than the present demand. The aim is to minimize transportation costs. Let d_{ij} be the time required to load one unit of product j on ship i, X_{ij} the number of units of j loaded on ship i, ai the loading time available on ship i, b_j the number of units of j which must be loaded, and C_{ij} the cost of assigning one unit of product j on ship i.

The problem is to find non-negative X_{ij} which satisfy the constrains

$\sum d_{ij} X_{ij} \pm X_{si} = a_i$,   where $a_i > 0$, i = 1, m

$\sum X_{ij} = b_j$,   where $b_j > 0$, j = 1,n

$X_{ij} \geq 0$

And minimize the cost

$$Z = \sum_{i}\sum_{j} C_{ij} X_{ij}$$

Once the optimal solution has been found, the sensitivity analysis and a parameter study can be performed. Consider a simple problem which optimal solution is given in Table 2.

Table 2: Optimal solution of a generalized transportation problem

		M_1		M_2		M_3		M_4		Slack	A_i	u_i
O_1	0.2 / 0.2	200	0.25 / 0.4	-0.15	0.35 / 0.25	-0.15	0.4 / 0.4	-0.1	0 / 1	260	300	0
O_2	0.6 / 0.8	-0.1	0.6 / 0.7	-0.5	0.2 / 0.3	600	0.3 / 0.4	1000	0 / 1	20	600	0
O_3	0.3 / 0.7	-0.1	0.1 / 0.3	300	0.5 / 0.6	-0.3	0.7 / 0.8	-0.4	0 / 1	310	400	0
B_j	200		300		600		1000					
v_j	0.2		0.1		0.2		0.3					

Table 2 shows the basic variables of the optimal solution are X_{11} = 200, X_{23} = 600, X_{24} = 100, X_{32} =300, X_{s1}= 230, X_{s2} =20, and X_{s3}=310. These values were obtained after to run a program designed by the author that uses the generalized transportation algorithm designed by Hadley [Hadley, 1963]. From the optimal basic solution are developed all analysis. These are

- Analysis of a C^b_{ij} associated with a basic variable.
- Analysis of a C_{ij} associated with a not basic variable.
- Analysis of a b_j associated with a requirement.

Analysis of a C_{ij} associated with a basic variable

Now, C_{23}^b = 0.2 was selected, and it will be perturbed. That is

$C^b_{23(new)} = C^b_{23(old)} * (1 + L)$

$C^b_{23(new)} = 0.2 (1 + L) = 0.2 + 0.2L$

This perturbation affects the shadow prices, so they will be calculated again, and the equation system ($d_{ij} u_i + v_j - C_{ij}$) will are in function of L.

$0.2 U_1 + V_1 = 0.2$

$0.3 U_2 + V_3 = 0.2(1 + L)$

$0.4 U_2 + V_4 = 0.3$

$0.3 U_3 + V_2 = 0.1$

$U_1 = 0$

$U_2 = 0$

$U_3 = 0$

$0.2 (0) + V_1 = 0.2 \rightarrow V_1 = 0.2$

$0.3 (0) + V_3 = 0.2 \rightarrow V_3 = 0.2(1 + L) = 0.2 + 0.2L$

$0.4 (0) + V_4 = 0.3 \rightarrow V_4 = 0.3$

$0.3 (0) + V_2 = 0.1 \rightarrow V_2 = 0.1$

Calculate **-** ($Z_{ij} - C_{ij}$) associated with not basic variables considered the value of L in each case.

$d_{12} U_1 + V_2 - C_{12} \rightarrow 0.4 (0) + 0.1 - 0.25 = -0.15$

$d_{13} U_1 + V_3 - C_{13} \rightarrow 0.25 (0) + 0.2 + 0.2L - 0.35$

$\rightarrow$ **L = 0.75**

$d_{14} U_1 + V_4 - C_{14} \rightarrow 0.4 (0) + 0.3 - 0.4 = -0.1$

$d_{21} U_2 + V_1 - C_{21} \rightarrow 0.8 (0) + 0.2 - 0.6 = -0.4$

$d_{22} U_2 + V_2 - C_{22} \rightarrow 0.7 (0) + 0.2 -0.6 = -0.4$

$d_{31} U_3 + V_1 - C_{31} \rightarrow 0.7 (0) + 0.2 - 0.3 = -0.1$

$d_{33} U_3 + V_3 - C_{33} \rightarrow 0.6 (0) + 0.2 + 0.2L - 0.5$

$\rightarrow$ **L = 1.5**

$d_{34} U_3 + V_4 - C_{34} \rightarrow 0.8 (0) + 0.3 - 0.7 = -0.4$

Find the minimum of positive L_s.

Minimum (0.75, 1.5) = 0.75

$C^b_{23(new)} = 0.2 (1 + L) = 0.2 + 0.2L$

$C^b_{23(new)} = 0.2 (1 + 0.75) = 0.2 + 0.15 = 0.35$

$C^b_{23min} \leq C^b_{23} \leq C^b_{23max}$

$0 \leq C^b_{23} \leq 0.35$

The optimal basic solution won't change if the values of the cost C^b_{23} are inside of interval [0.2, 0.35]. This result can be verified if the shadow prices are calculated again.

Analysis of a C_{ij} associated with a not basic variable

Now, the cost C_{31} = 0.3 associated with the not basic variable X_{31} will be perturbed. That is

$Z_{31} - C_{31} \leq 0$

$Z_{31} - C_{31} = d_{31}\, u_3 + v_1 - C_{31}$

$Z_{31} - C_{31} = d_{31}\, u_3 + v_1 - (C_{31} + \Delta C_{31}) \leq 0$

$\Delta C_{31} \geq d_{31}\, u_3 + v_1 - C_{31}$

$\Delta C_{31} \geq 0.7\,(0) + 0.2 - 0.3$

$\Delta C_{31} \geq 0.2 - 0.3$

$\Delta C_{31} \geq -0.1$

$C_{31(new)} = 0.3 - 0.1 = 0.2$

$0.2 \leq C_{31} \leq {}_{+}\infty$

The optimal basic solution won't change if the values of the cost C_{31} are inside of interval $[0.2, {}_{+}\infty]$. While the value of C_{31} is incremented the optimality condition to the optimal feasible solution will not change. However, the value of C_{31} can be decremented until 0.2. Values under this amount will change the optimality condition to this solution, and the variable associated with this cost could enter into the basis.

Analysis of a b_j associated with a requirement

The requirement b_1 is selected because the cell that corresponds with the basic variable X_{11} is on a row with a slack variable different from zero. Now, b_1 will be perturbed. That is,

$b_{1(max)} = b_{1(old)}\,(1 + L) = 200\,(1 + L)$

The critical value of L that reduces to zero the corresponding slack on the selected row will be calculated.

$d_{ij}\, X_{ij}\,(1 + L) = a_i$
$0.2 * 200 * (1 + L) = 300$
$40 + 40\,L = 300$

$L = (300 - 40) / 40 = 6.5$

$b_{1(max)} = 200\,(1 + L) = 200\,(1 + 6.5)$
$b_{1(max)} = 1500$

$200 \leq b_1 \leq 1500$

The requirement b_1 could be incremented to 1500 from its initial value 200. It is possible because of the surplus variable X_{s1} in the optimal basic solution.

Summary

With the objective of attenuating indetermination and to facilitate decision-making we developed a novel method to simulate the behavior of real systems which structures are analogous to generalized transportation problem. The method simulates changes on the parameters for the optimal solution of the mathematical model. The proposed method focuses on analyzing requirements and cost associated with variables (basic and not basic). Modifications made to this algorithm later permitted us to obtain a version adapted to standard transportation problem. In both cases, the software prototype was built.

The algorithm described has solved different problems analogous to production allocation and distribution, and constitute an interesting teaching and learning tool to students at the university level.

References

Gil, D. (2003). Sensitivity Analysis and Parameter Study of the Generalized Transportation Problem. Proceedings of EURO/INFORMS, Istanbul Turkey, July 6-11, 2003.

Garcia, D., Guillen, E. and Garcia Ma., (1993). Load Transportation Balancing Modeling. Industrial Engineering Journal, Vol. XIV, No. 1, ISSN 0258-5960, Cuba. 63-67

Garcia, D et al, (1991). Load Transportation Balancing Modeling to Havana Port. Research Report. Computer science Department, Havana Polytechnic Institute Cuba, 1-25.

Garcia, D and Garay, M., (1988). Parametric Analysis of Generalized Transportation Problem" Industrial Engineering Journal Vol. IX, No. 1, ISSN 0258-5960, Cuba 79 – 83.

Garcia, D. and Garay, M. (1987). Generalized Transportation Problem: Its algorithms and computer program. Industrial Engineering Journal Vol. VIII, No. 2, Cuba 153-158

Hadley, G. (1963). Linear Programming, Addison-Wesley Publishing Company, Inc. Second Edition. U.S.A. 316-322, 273

Robotic algorithms for mowing a field

Rex Oleson II[1,2,3], D. J. Kaup[1,2,4], Neil Finkelstein[5]

[1] University of Central Florida, Department of Mathematics
P.O. Box 161364, Orlando, FL 32816-1364
[2] University of Central Florida, Institute For Simulation and Training
3280 Progress Drive, Orlando, FL 32826
[5]Simulation Technology Center
12423 Research Parkway, Orlando, FL, 32826-3276
[3]quazerin@juno.com,[4]kaup@mail.ucf.edu,[5]neal.finkelstein@us.army.mil

Abstract

We construct two robotic algorithms for the simulation of N robots mowing a field, such that no patch in the field will be left un-mowed by the robots. The first algorithm is a wall following algorithm, by which the robots follow the exterior walls, and circle in towards the center of the field. The second algorithm is one whereby the robots go up and down, sweeping from East to West, across the field. In order to evaluate the efficiency of each algorithm in directing robots to mow a field, we simulate the two algorithms in three different scenarios: 1) a simple open square field, 2) a square field with two obstacles on the borders, and 3) a square field with two obstacles on the borders, and an open obstacle in the center. The number of robots is varied in the simulations from one to ten, while the field to be mowed is taken to be a square field, consisting of 100 elements by 100 elements. We also define a cost function for measuring the efficiency of each robot in mowing a field. As we will demonstrate, each algorithm is found to be successful in directing the robots in mowing the field, in directing the robots to avoid the obstacles, and without leaving any elements un-mowed. Also, from our definition of the cost function, we will demonstrate that there are no serious differences in the efficiency of each algorithm.

Introduction

The purpose of the paper is to devise methods for directing a robot, or robots, in mowing a field. We will construct algorithms that will direct a group of robots in mowing a field. Two separate algorithms will be created. Simulations for these two algorithms will be done, over three different scenarios, to validate that they accomplish the goal of mowing the field. The data collected from the simulations will be discussed, as well as problems associated with the applications of the algorithms.

Model

Field Definition

The first aspect of the model is to define the field to be mown. Then we discuss a method of breaking up the field into elements to be mowed, and finally, we define how we will describe the state of each element of the field.

The simplest model to use for the field to be mowed is to embed it into a square of size $X_a \text{ x } Y_a$, where $X_a = Y_a$, which is then divided into square elements of size $d_s \text{ x } d_s$. We then can refer to the individual elements of the square by their position on the grid as a (x, y) coordinate.

When one is dividing the field up into the individual elements, two factors must be considered. First is the mowing width of the robot. The d_s element size must be taken sufficiently small to guarantee that the robot will mow the entire element as it passes over and through the element. The second condition is the issue of how close one can come to any obstructions in the field. If the squares are made too large, then around and adjacent to any obstruction in the field, the robot could leave un-mown areas. Thus one needs to have the individual elements sufficiently small to avoid this.

There are three states that a given element in the field could be found in: OPEN, MOWED, or CLOSED. OPEN represents an un-MOWED element of the area, MOWED represents a MOWED element of the area, and CLOSED represents an obstacle in that element, which prevents the robot from entering it. An OPEN element becomes a MOWED element once a robot has entered the element. The function $f(x,y,t)$, will be the world function, representing the state of the (x,y) element at time *t*. It will take on values as:

$$f(x,y,t) = \begin{cases} 0 & \text{if the element is OPEN,} \\ 1 & \text{if the element has been MOWED,} \\ 2 & \text{if the element is CLOSED.} \end{cases} \quad (1)$$

The field is considered to be a confined space. Thus all elements outside of the field are CLOSED

Algorithms

An algorithm will define how a robot will move from one element of the field to the next element. There are some general aspects that are common to both algorithms defined. The first algorithm is a Border-Following algorithm. The second algorithm is an Up-Down algorithm. The common aspects of the two algorithms will be defined first, and then the individual implementations of each will be covered.

Common Aspects. We will introduce a series of functions to determine, from the state of any element in the field, where we will next send the robot. The general model being used for the movement of the robot will be described, along with defining the direction that the robot is facing. In the case where the robot ends up being surrounded by MOWED, and CLOSED elements, we need a way to move it in towards OPEN elements. For that purpose, we define an attractive/repulsive force that is used by both algorithms, causing the robot to go towards OPEN elements and away from CLOSED or MOWED elements.

All these forces are based upon the state of the elements in the field. To calculate them, we first define a Boolean set of functions, which act on the world function, to determine the state of an element:

$$g(f) = \begin{cases} 1 & \text{if } f = 2, \\ 0 & \text{elsewhere.} \end{cases} \quad (2)$$

$$h(f) = \begin{cases} 1 & \text{if } f = 1, \\ 1 & \text{if } f = 2, \\ 0 & \text{elsewhere.} \end{cases} \quad (3)$$

$$i(f) = \begin{cases} 1 & \text{if } f = 1, \\ 0 & \text{elsewhere.} \end{cases} \quad (4)$$

$$j(f) = \begin{cases} 1 & \text{if } f = 0, \\ 0 & \text{elsewhere.} \end{cases} \quad (5)$$

The function *g* is nonzero if the state given is CLOSED. The function *h* is nonzero if the state given is either MOWED or CLOSED, while *i* or *j* is nonzero if the state of an element is MOWED or OPEN, respectively.

The method of directing the robot motion is done by an angle, ϕ, and a distance of movement, r. The distance of movement r is confined to 1. This way, each time the robot moves, it will only move one element at a time. We also restrict the motion to be such that the final value of ϕ must be a multiple of $\frac{\pi}{2}$. This means that the robot will only be allowed to move to the four elements surrounding, and adjacent to it. In determining the angle to which the robot is to move, we first determine a general, continuous vector, *m*, in which one would like the robot to go. This vector will be a function of the angle that the robot is facing, ϕ, and will have x and y components, denoted as m_x, and m_y respectively.

To determine ϕ at the new time step, we take the values of m_x and m_y, and find the angle (within $\frac{\pi}{2}$) closest to *m*. This is done by means of:

$$\phi(t+\Delta t) = \left[round\left(\frac{\arctan \frac{m_x}{m_y}}{\frac{\pi}{2}} \right) \bmod 4 \right] * \left(\frac{\pi}{2} \right) \quad (6)$$

The definition of mod is that used by Gallian in "Contemporary Abstract Algebra" (Gallian 1994). The function "round" truncates the variable if it lies between the integer value of the variable and is less than the integer value+0.5, while it rounds up if the value is the integer+0.5 and less than the next integer value.

The neighboring elements, which will directly influence the robot, through the algorithms, are the eight elements directly adjacent to the element where the robot is. To facilitate looking at these elements, based on the direction the robot is facing, ϕ, a set of functions are defined. We extend the world function, defined in Equation 1, for the elements to the forward, forward right, right, back right, back, back left, left and forward left of a robot as the functions f_f, f_{fr}, f_r, f_{br}, f_b, f_{bl}, f_l and f_{fl} respectively (Oleson 2003).

A common component to both algorithms is an attractive/repulsive set of forces, dependent on all the elements of the field. The use of these attractive/repulsive forces is similar to the Artificial Potential Field(APF) approach in traditional path finding (Vadakkepat 2001). The APF method gives an attractive force towards any goal element, and a repulsion force from any obstacle. In this setting, each OPEN element in the field contributes to an attractive force on the robot, as if it were a goal point, and any CLOSED, or MOWED elements adjacent to the robot imparts a repulsive force, acting as an obstacle. These attractive and repulsive forces are defined by:

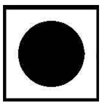
$$Att_c\left(X_i,Y_i,t\right)=\left[\sum_{(x,y):(x,y)\neq(X_i,Y_i)}\left(\frac{j(f(x,y,t))*(X_i-x)}{((X_i-x)^2+(Y_i-y)^2)^{\frac{5}{2}}}\right)\right] \quad (7)$$
$$+h(f(X_i-1,Y_i,t))-h(f(X_i+1,Y_i,t))$$

$$Att_y\left(X_i,Y_i,t\right)=\left[\sum_{(x,y):(x,y)\neq(X_i,Y_i)}\left(\frac{j(f(x,y,t))*(Y_i-y)}{((X_i-x)^2+(Y_i-y)^2)^{\frac{5}{2}}}\right)\right] \quad (8)$$
$$+h(f(X_i,Y-1_i,t))-h(f(X_i,Y+1_i,t))$$

A drop-off in the attractive force, as the distance from the robot increases, is required so that the attractive forces do not overpower the directional forces, applied by the individual algorithms. The $\frac{5}{2}$'s power in the denominator was found experimentally. This value had to be sufficiently large to create a rapid drop-off, but small enough to yet pull a robot towards any OPEN elements. The main reason for these attractive forces is to give the robot a gentle nudge towards any un-MOWED elements of the field, when it is surrounded by CLOSED or MOWED elements. If this force were too large, then the attraction and repulsion forces would overpower the forces applied by the rest of the algorithm.

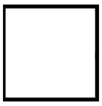
To display the two algorithms, a series of diagrams are used to describe the forces acting on the robots. The diagrams will display a robot at the center of a 3x3 grid of elements, representing the square the robot is on, with the eight surrounding elements. There are two different representations of the robot.

represents a robot where the direction is important. The arrow points in the direction the robot is facing, and the squares around it are represented as forward, back, right, and left.

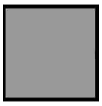


represents a robot where the direction that the robot is facing is not important. The surrounding squares are represented as North, South, East, and West.

The squares immediately surrounding the robot are shaded to represent the state of that element. The shadings are defined as follows



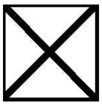
represents an element which is Open.



represents an element which has been Mowed.

represents an element which is Closed.

represents an element which is Closed or Mown.

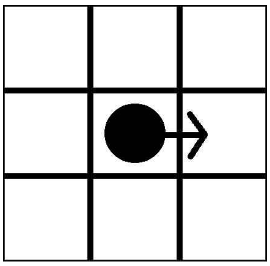


represents an element whose state is immaterial.

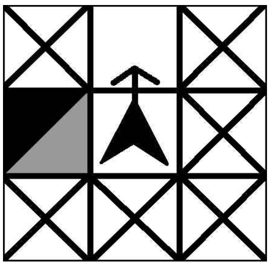
Border-Following algorithm. The first algorithm follows the boundary wall and spirals in towards the center of the field, until all of the OPEN elements have been MOWED. The first thing the robot does, if it is not against a wall, is to head East until it reaches a wall. After reaching the wall, it turns and follows the wall. Once reaching another wall, it turns 90° again and follows the wall. This process continues until the robot reaches an element that has been MOWED. It will then turn 90°, and follow the MOWED elements. The robot will continue the previous mowing techniques along the MOWED areas, continuing to spiral in toward the center of the field.

The implementation of the algorithm requires the consideration of additional specific cases, in order to ensure that the robot is following the outer perimeter, as it mows the field.

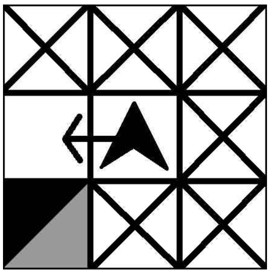
$m_{a1} =$



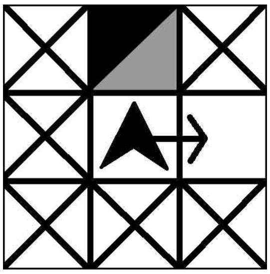
$m_{a2} =$



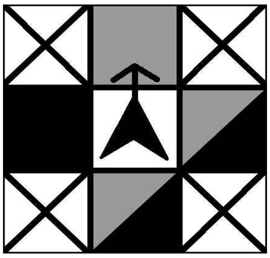
$m_{a3} =$



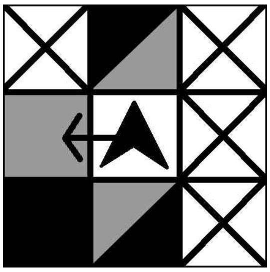
$m_{a4} =$



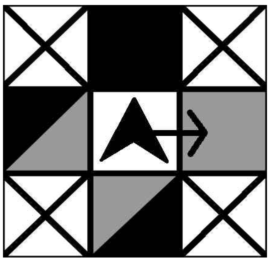
$m_{a5} =$



$m_{a6} =$



$m_{a7} =$



Figures 1-7: These Figures represent the influences, for the Border-Following algorithm, that cause the desired outcome based on the surrounding elements of the robot and the direction it is currently facing. The arrow shows the square to which the robot will next move to.

Now we need to separate these into X and Y directional components, based on Figures 1-7. So we end up with:

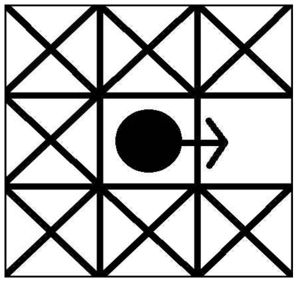
$$m_{xa}(m_a, X_{ia}, Y_{ia}, t + \Delta t) = Att_{xa} + m_{xa1} + m_{xa2} + m_{xa3} + m_{xa4} + m_{xa5} + m_{xa6} + m_{xa7} \quad (9)$$

$$m_{ya}(m_a, X_{ia}, Y_{ia}, t + \Delta t) = Att_{ya} + m_{ya1} + m_{ya2} + m_{ya3} + m_{ya4} + m_{ya5} + m_{ya6} + m_{ya7} \quad (10)$$

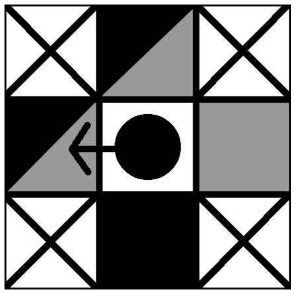
Up-Down Algorithm. The second algorithm is one where the robot goes up and down, sweeping from East to West across the field. The robot starts by heading East until it reaches a wall, and then goes up along the wall. Once it reaches the upper wall, the robot steps one element to the West and then goes South until it reaches a wall. Once the wall is reached, the robot steps one element to the West, and then goes North. The robot continues this pattern until all elements of the field have been MOWED.

This pattern does not take into consideration the direction that the robot is facing. The important aspects of this pattern are only what the elements are surrounding the robot, and not the direction that the robot is facing.

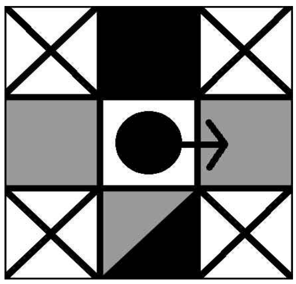
$m_{b1} =$



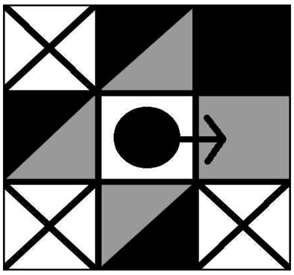
$m_{b2} =$



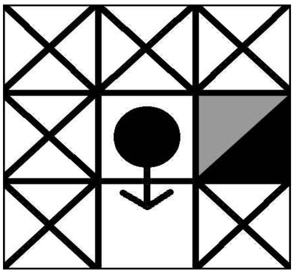
$m_{b3} =$



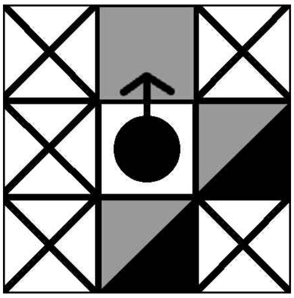
$m_{b4} =$



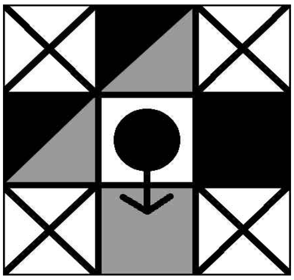
$m_{b5} =$



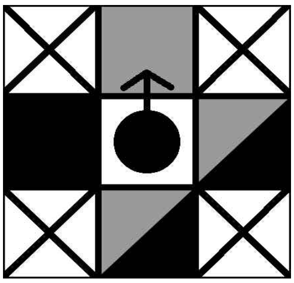
$m_{b6} =$



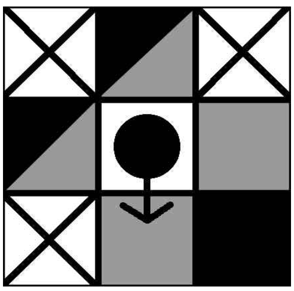
$m_{b7} =$



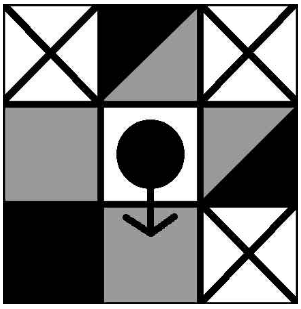
$m_{b8} =$



$m_{b9} =$



$m_{b10} =$



Figures 8-17: These Figures represent the directions to the robot, for the Up-Down Algorithm, based on the elements surrounding the robot and the direction it is currently facing. The arrow shows the square to which the robot will move to.

Now we need to separate these into X and Y directional components, based on Figures 8-17. So we end up with:

$$m_{xb}(m_b, X_{ib}, Y_{ib}, t+\Delta t) = Att_{xb} + m_{xb1} + m_{xb2} + m_{xb3} + m_{xb4} + m_{xb5} + m_{xb6} + m_{xb7} + m_{xb8} + m_{xb9} + m_{xb10} \quad (11)$$

$$m_{yb}(m_b, X_{ib}, Y_{ib}, t+\Delta t) = Att_{yb} + m_{yb1} + m_{yb2} + m_{yb3} + m_{yb4} + m_{yb5} + m_{yb6} + m_{yb7} + m_{yb8} + m_{yb9} + m_{yb10} \quad (12)$$

Simulation

The algorithms were tested through a series of simulations. The simulations were run for three different scenarios, and then the algorithms were compared according to the defined cost function, Equation 16. The three different scenarios are an open area, an area with two obstacles on the boundaries, and an area with two obstacles on the boundaries and a complicated gated obstacle near the center.

The first simulation is with the entire field open, no obstacles anywhere in the field. This gives us a clean, but basic look, at how each algorithm covers the field. This also allowed for a visual check to ensure that the robot was covering the field by the manner describe in the algorithm.

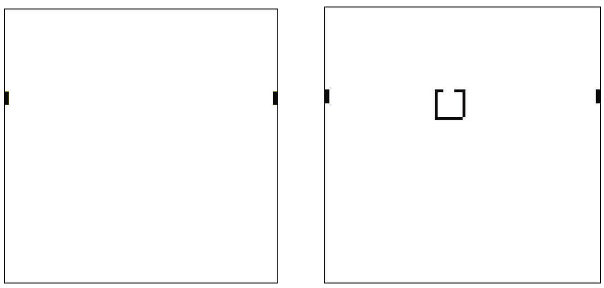


Figure 18-19: Figure 18 (on the left) demonstrates the Field with only boundary objects. Figure 19 (on the right) demonstrates the boundary objects and the central obstacle.

The second simulation is with obstacles on the east and west walls. These obstacles are identical except for being on opposite walls, as displayed in Figure 18. This gives a look at how the algorithms work on non-square fields, and gives a basic look at its avoidance of obstacles.

The third simulation is the same as the second simulation but with an added small gated area near the center, as displayed in Figure 19. This gated area is rectangular in shape with a small opening in the northern wall. There is also a small corner piece deliberately left open in the obstacle at the southeast corner. This is a complicated shape, separating the field into an exterior field, and an interior one. This last simulation demonstrates the ability of the algorithm to compensate for obstacles in the field, along with demonstrating how well the algorithms will go to OPEN areas in the field.

Goals and Costs

Two algorithms have been defined, and simulations were run to compare them. The aspects that we now compare are 1) did the algorithms accomplish the goal, and 2) if they did, how long did it take. Also while accomplishing the goal, what other costs did the robot accumulate? The time for covering the area and a few reasonable costs are used to define a cost function, which is used to compare the two algorithms.

Goal definition

The Goal, as previously stated, is to mow/cover the entire field. Therefore the first and the most important aspect of comparing the algorithms would be the time it takes to mow the field, represented as:

$$T_a = \text{Total Time to cover area} \quad (13)$$

Costs

Costs of the algorithms are less tangible than the goals. One such cost is the number of times that we move over an element of the field, more then once. This is important to us, since we can easily see that the more times we multiply cross a given element, the longer it will take us to mow the whole field. This is therefore included in the previously defined variable T_a, Equation 13. This is an important aspect that would need to be reduced, in order to optimize any of the algorithms. The number of crossovers is defined as:

$$C_o = \text{number of crossovers} \quad (14)$$

Another cost is the number of turns taken to complete the mowing of the field. This number is important because the turning of a robot can be a time and energy consuming process. This is recorded by noting the number of times that the robot changes the direction it is facing, and is defined as:

$$T_u = \text{number of turns} \quad (15)$$

Cost Function

From the goal, Equation 13, along with the other costs, Equation 14 - 15, an overall cost function can be defined, as a linear combination of these individual costs:

$$Cost = T_a + C_o + \frac{1}{2} T_u \quad (16)$$

The cost gives us a way of comparing algorithms, and will also allow us to judge any optimization of the algorithms. The weights associated with each goal/cost create a ranking system for the goals/costs in comparison to each other. In the above, Equation 16, we have taken the time for overall mowing to be most important, and the number of crossovers are the next most important factor in the choice of a better algorithm, while the number of turns is judged to be only about half as expensive as another crossover.

RESULTS

Let us compare the two algorithms by the cost function, Equation 16. As shown in Figure 20, we see that there is no major difference between the two algorithms. The Border-Following algorithm is generally better than the Up-Down algorithm in all cases, but only slightly.

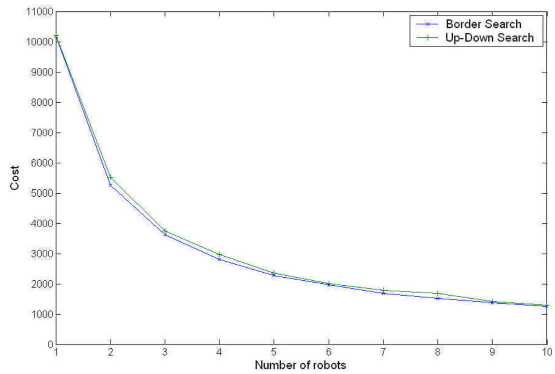


Figure 20: Cost function for the two algorithms.

Figure 21 shows the difference in cost and total time, which demonstrate much greater differences, due to the wide fluctuations in the number of turns and crossovers that can occur. Note that at the six-robot point, these two costs are rather close for each algorithm.

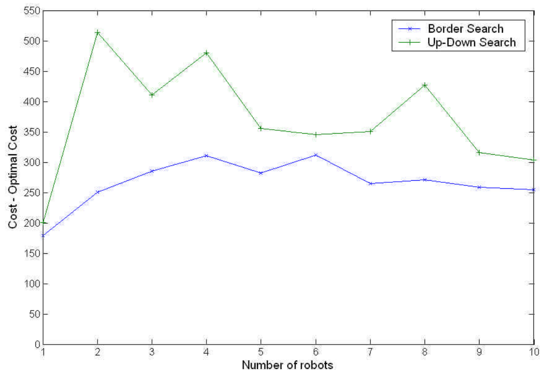


Figure 21:Comparison of the difference between the cost and the total time for both algorithms.

Now we consider the effects of placing obstacles in the field, as in Fig. 18. As seen in Fig. 22, the two algorithms are again in close agreement in most cases, with the Border-Following algorithm being slightly better. The Border-Following algorithm does slightly better in the three-robot case, but falls back towards the graph of the Up-Down algorithm. Also, we note that the Border-Following algorithm failed to cover the entire area in the ten-robot case, which is why no data point exists for it.

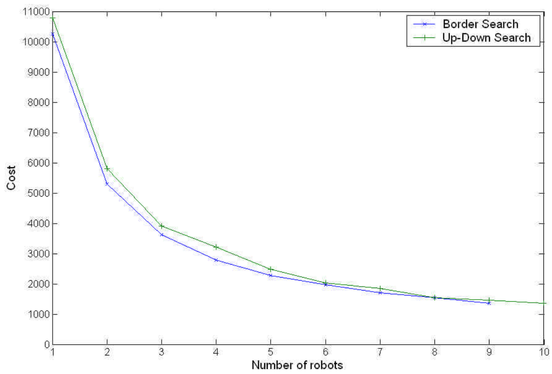


Figure 22:Comparison of costs for the field shown in Fig. 18.

Figure 23 is again the difference in the cost and total time, but for the field in Fig. 18. It shows that the two graphs approach each other as the number or robots increases. The difference for the Border-Following algorithm hardly changes as we increase the number of robots, while the difference for the Up-Down algorithm generally decreases. They reach approximately the same value for the eight-robot situation.

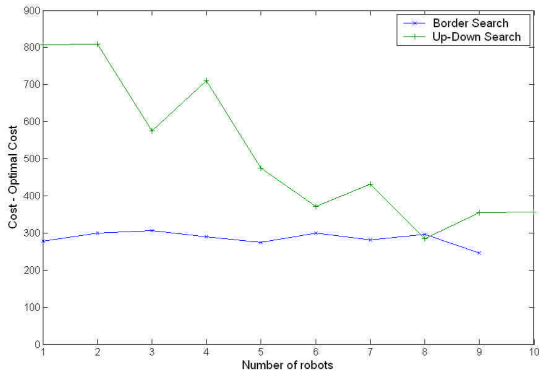


Figure 23:Comparison of the difference in the cost and total time, for the field in Fig. 18.

When we include a central obstacle in the field, as in Fig. 19, then the Border-Following algorithm still comes out to be slightly better than the Up-Down algorithm, but only in those simulations wherein it was able to complete the goal, as seen in Fig. 24. Results for the difference in the cost and total time are even more variable in this case, as one can see from Fig. 25. However, the same general principle seems to apply, in that the Up-Down-Following algorithm will complete the job always, but not as efficiency as the Border-Following algorithm. However, sometimes the latter failed to complete to job.

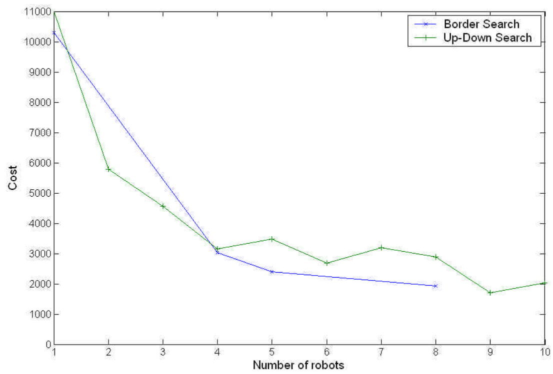


Figure 24:Comparison of the costs for the mowing of the field in Fig. 19.

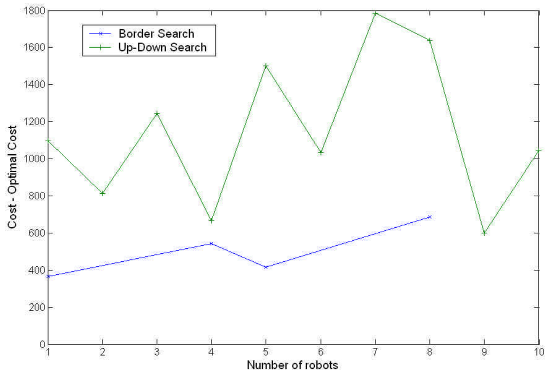


Figure 25: Comparison of the differences in the cost and total time for the field shown in Fig. 19.

CONCLUSION

Both algorithms, Border-Following and Up-Down, accomplished the goals of mowing the entire area when dealing with a single robot. Consequently, defining an algorithm as a set of difference equations is a reasonable approach. The simulations with different scenarios and multiple robots have been used to test the algorithms.

In general the Border-Following algorithm is faster, and more cost effective then the Up-Down algorithm. However, due to some cooperative problems with multiple robots in the Border-Following algorithm, the Up-Down algorithms might be a better choice since it always worked in even the complicated scenarios, for all number of robots tested. On the other hand, when both did complete the job, there was no major difference between the two algorithms, as seen in Figures 20-25. The general conclusion is that any reasonable algorithm will end up being comparable to any other reasonable algorithm, in terms of costs.

The key factors to consider when looking for an algorithm will be 1)how well they hold up in multiple robot situations, and 2) how well they deal with obstacles that are in the field.

References

Michael H. Bowling and Manuela~M. Veloso, Motion control in dynamic multi-robot enviroments, *RoboCup*, 1999, pp. 222-230.

Mark Brockington, Pawn captures wyvern: How computer chess can improve your pathfinding, *2000 Game Developer's Conference Proceedings*, 2000.

Roger Burkhart, Schedules of activity in the swarm simulation system., *OOPSLA '97 Workshop on OO Behavioral Semantics*, 1997.

Peter Fletcher, Hughes Hoyle, and C.~Wayne Patty, *Discrete mathematics*, PWS-Kent Publishing Company, 1991.

Joseph A. *Gallian, Contemporary abstract algebra*, D.C. Heath Company, 1994.

Otomar Hajek, *Pursuit games*, Academic Press, Inc., 1975.

Woong-Gie Han, Seung-Min Baek, and Tac-Young Kuc, Ga based on-line path planningof mobile robots playing soccer games, *Proceedings of the 1997 IEEE International Conference on Robotics & Automation*, 1997, pp. 522-525.

Rex R. Oleson II, Mathematical robotic algorithms for mowing a field, Master's thesis, University of Central Florida, 2003.

Rufus Isaacs, *Differential games*, John Wiley and Sons, Inc., 1967.

T. H. Lee, H. K. Lam, F. H. F. Leung, and P. K. S. Tam, A path planning method for micro robot soccer game, *Proceedings of the 27th Annual Conference of the IEEE Industrial Electronics Society, 2001*, pp. 626--629.

R. Duncan Luce and Howard Raiffa, *Games and decisions*, Dover, 1985.

Takayuki Nakamura, Development of self-learning vision-based mobile robots for acquiring soccer robots behaviours, *Proceedings of the 1998 IEEE International Conference on Robotics & Automation*, 1998, pp. 2592--2598.

Leon A. Petrosjan and Nikolay~A. Zenkevich, *Game theory*, World Scientific, 1996.

Craig Reynolds, Flocks, herds, schools: A distributed behavioral model, *Proceedings of the 1987 SIGGRAPH Conference*, 1987, pp. 25--34.

Craig W. Reynolds, Steering behaviors for autonomous characters, *Game Developers Conference 1999*, 1999.

Prahlad Vadakkepat, Tong~Heng Lee, and Liu Xin, Application of evolutionary artificial potential field in robot soccer system, *Proceedings of the 9th IFSA World Congress and 20th NAFIPS International Conference*, 2001, pp. 2781-2785.

Ching-Chang Wong, Ming-Fong Chou, Chin-Po Hwang, Cheng-Hsin Tsai, and Shys-Rong Shyu, A method for obstacle avoidance and shooting action of the robot soccer, *Proceedings of the 2001 IEEE International Conference on Robotics & Automation*, 2001, pp. 3778-3782.

Byoung-Tak Zhang and Sung-Hoon Kim, An evolutionary method for active learning of mobile robot path planning, *8th Int. Conf. Advanced Robotics*, 1997, pp.~312--317.

Non-Quantified Modeling with Computer-Aided Morphological Analysis

Tom Ritchey

Swedish Defense Research Agency
SE-172 90 Stockholm, Sweden
ritchey@foi.se

Abstract

Morphological analysis (MA), pioneered by Fritz Zwicky at the California Institute of Technology in the 1930s and 40s, was developed as a method for structuring and investigating the total set of relationships contained in multi-dimensional, non-quantifiable, problem complexes. During the past ten years, MA has been extended, computerized and applied in developing futures scenarios, structuring and analyzing complex policy spaces, and modeling strategy alternatives. This article outlines the fundamentals of the morphological approach and describes recent applications in strategy modeling.

Introduction

Morphological analysis (MA) was developed by Fritz Zwicky – the Swiss-American astrophysicist and aerospace scientist based at the California Institute of Technology (CalTech) – as a method for structuring and investigating the total set of relationships contained in multi-dimensional, non-quantifiable, problem complexes (Zwicky 1966, 1969).

Zwicky applied this method to such diverse tasks as the classification of astrophysical objects, the development of jet and rocket propulsion systems, and the legal aspects of space travel. More recently, morphological analysis has been extended and applied by a number of researchers in the U.S.A and Europe in the field of futures studies, policy analysis and strategy modeling (Rhyne 1981, 1995; Coyle *et.al.* 1994, 1995; Jenkins 1997; Ritchey 1997, 2003; Stenström & Ritchey 1999; Eriksson & Ritchey 2000). The method is currently experiencing somewhat of a renaissance, not the least because of the development of small, fast computers and flexible graphic interfaces.

This paper will begin with a discussion of some of the methodological problems confronting complex, non-quantified systems modeling as applied to futures studies and strategy analysis. This is followed by a presentation of the fundamentals of the morphological approach along with two recent applications: the development of an instrument for evaluating preparedness for accidents or terrorist actions involving chemical releases; and a tactical scenaro laboratory for evaluating requirements for future artillery systems for the Army Tactical Command.

Methodological background

Developing futures scenarios and modeling complex socio-technical and organization systems presents us with a number of difficult methodological problems. Firstly, many of the factors involved are not meaningfully quantifiable, since they contain strong social, political and organizational dimensions. This means that traditional quantitative methods, mathematical modeling and simulation are relatively useless.

Secondly, the uncertainties inherent in such problem complexes are in principle non-reducible, and often cannot be fully described or delineated. This includes both antagonistic uncertainty (conscious, willful actions among actors) and so-called non-specified uncertainty (for instance, uncertainties concerning what types of scientific and technological discoveries will be made in the future). This represents even a greater blow to the idea of causal modeling and simulation.

Finally, the creative process by which conclusions are drawn in such studies is often difficult to "trace" – i.e. we seldom have an adequate "audit trail" describing the iterative process from problem formulation, through alternative generation to specific solutions or conclusions. Without some form of traceability we have little possibility of scientific control over results, let alone reproducibility.

An alternative to mathematical and causal modeling is a form of non-quantified modeling relying on ”judgmental processes” and internal consistency, rather than causality. Causal modeling, when applicable, can – and should – be used as an aid to judgment. However, at a certain level of complexity (e.g. at the social, political and cognitive level), judgment must often be used, and worked with, more or less directly. The question is: How can judgmental processes be put on a sound methodological basis?

Historically, scientific knowledge develops through cycles of analysis and synthesis: every synthesis is built upon the results of a proceeding analysis, and every analysis requires a subsequent synthesis in order to verify and correct its results (Ritchey, 1991). However, analysis and synthesis – as basic scientific methods – say nothing about a problem having to be quantifiable.

Complex social-political systems and policy fields can be analyzed into any number of non-quantified variables and ranges of conditions. Similarly, sets of non-quantified conditions can be synthesized into well-defined relationships or configurations, which represent “solution spaces”. In this context, there is no fundamental difference between quantified and non-quantified modeling.

Morphological analysis – extended by the technique of internal "cross consistency assessment" (CCA, see below) – is a method for rigorously structuring and investigating the internal properties of inherently non-quantifiable problem complexes, which contain any number of disparate parameters. It encourages the investigation of boundary conditions and it virtually compels practitioners to examine numbers of contrasting configurations and policy solutions. Finally, although judgmental processes may never be fully traceable in the way, for example, a mathematician formally derives a proof, MA goes a long way in providing as good an audit trail as one can hope for.

The morphological approach

Essentially, morphological analysis is a method for identifying and investigating the total set of possible relationships contained in any given, multi-dimensional problem complex that can be parameterized. morphological analysis.

The method thus begins by identifying and defining the most important dimensions (or parameters) of the problem complex to be investigated, and assigning each parameter a range of relevant “values” or conditions. This is done in natural language. A morphological field – also fittingly known as a “Zwicky box” – is constructed by setting the parameters against each other in an n-dimensional configuration space (see Figure 1, below). Each configuration contains one particular ”value” or condition from *each* of the parameters, and thus marks out a particular state or (formal) solution within the problem complex.

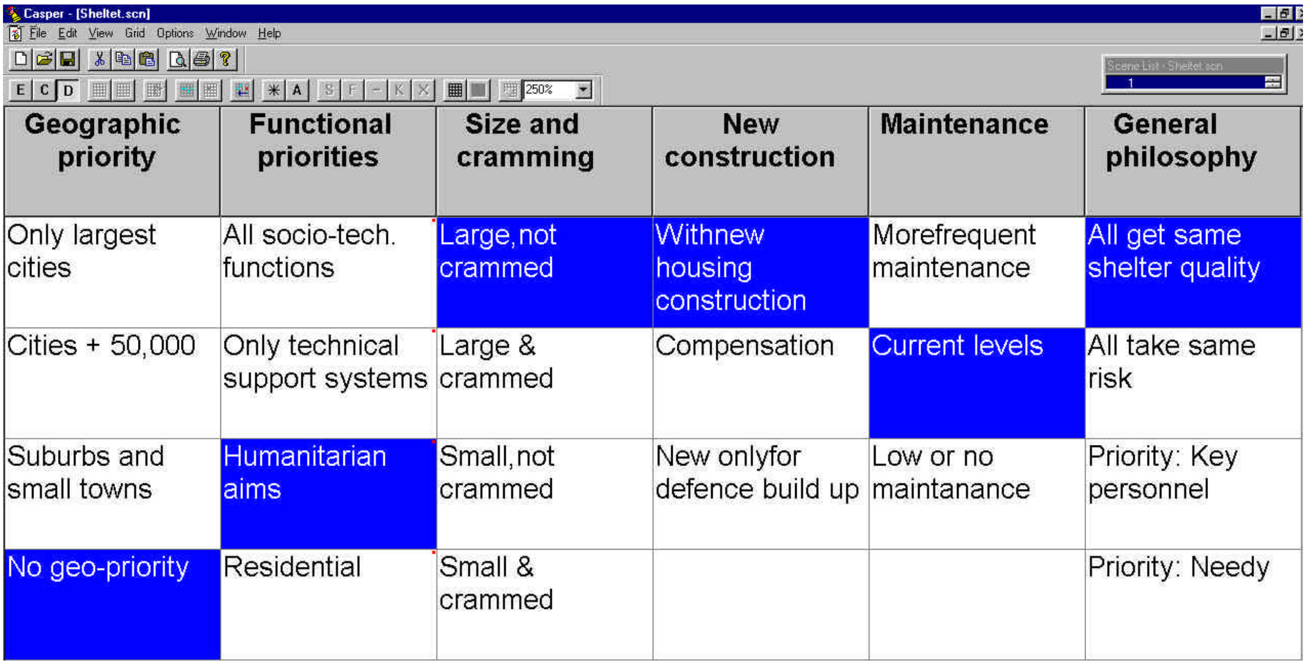
For example, suppose that we wished to review and re-conceptualize the Swedish national bomb shelter program since the end of the cold war (a task we actually did for the Swedish Rescue Services Board in the middle of the 1990's). This is a complex policy issue, which contains a number of disparate dimensions or variables: financial, technical, political, geographical, ethical, etc. As an example, consider the following dimensions:

1. What size of cities should we concentrate on?
2. Who should we be sheltering?
3. How big (or small) and how many people should we pack into such shelters?
4. Under what circumstances should we take up new construction?
5. What level of maintenance?
6. What is the point (philosophy) of the shelter program?

With these 6 dimensions, a morphological field like the one shown in Figure 1 could be created.

If a morphological field is small enough, one can examine all of the configurations in the field, in order to establish which of them are possible, viable, practical, interesting, etc., and which are not. In doing so, we mark out in the field a relevant “solution space". The “solution space” of a Zwickian morphological field consists of the subset of configurations, which satisfy some criteria -- usually the criteria of internal consistency.

However, a typical morphological field may contain between 50 and 500 thousand configurations, far too many to inspect by hand. Thus the next step in the analysis-synthesis process is to examine the internal relationships between the field parameters and "reduce" the field by weeding out all mutually contradictory conditions.



Geographic priority	Functional priorities	Size and cramming	New construction	Maintenance	General philosophy
Only largest cities	All socio-tech. functions	Large,not crammed	Withnew housing construction	Morefrequent maintenance	All get same shelter quality
Cities + 50,000	Only technical support systems	Large & crammed	Compensation	Current levels	All take same risk
Suburbs and small towns	Humanitarian aims	Small,not crammed	New onlyfor defence build up	Low or no maintanance	Priority: Key personnel
No geo-priority	Residential	Small & crammed			Priority: Needy

Figure 1: Segment of a morphological field used in the Swedish bomb shelter study. This field contains 4x4x4x3x3x4 (=2304) possible "configurations" -- one of which is highlighted.

This is achieved by a process of cross-consistency assessment: all of the parameter values in the morphological field are compared with one another, pair-wise, in the manner of a cross-impact matrix (see Figure 2, following page). As each pair of conditions is examined, a judgment is made as to whether – or to what extent – the pair can coexist, i.e. represent consistent relationship. Note that there is no reference here to causality, but only to internal consistency.

There are two types of inconsistencies involved here: purely logical contradictions (i.e. those based on the nature of the concepts involved); and empirical constraints (i.e. relationships judged be highly improbable or implausible on empirical grounds). Normative constraints can also be applied, although these must be used sparingly, and one must be very careful not to allow prejudice to rule such judgments.

This technique of using pair-wise consistency relationships between conditions, in order to weed out internally inconsistent configurations, is made possible by a principle of dimensionally inherent in the morphological approach. While the number of configurations in a morphological field grows exponentially with each new parameter, the number of *pair-wise relationships between conditions* grows in proportion to the triangular number series -- a quadratic polynomial. Naturally, there are practical limits reached even with quadratic growth. The point, however, is that a morphological field involving as many as 100,000 formal configurations can require no more than few hundred pair-wise evaluations in order to create a solution space.

When this solution space (or outcome space) is synthesized, the resultant morphological field becomes a flexible model, in which anything can be "input" and anything "output". Thus, with computer support, the field can be turned into a laboratory with which one can designate one or more variables as inputs, in order to examine outputs or solution alternatives (see Figures 3 and 4, below).

The morphological approach has several advantages over less structured approaches. Zwicky called MA "totality research" which, in an "unbiased way attempts to derive all the solutions of any given problem". It seeks to be integrative and to help discover new relationships or configurations, which might be overlooked in other – less structured – methods. Importantly, it encourages the identification and investigation of boundary conditions, i.e. the limits and extremes of different parameters within the problem space. For Zwicky, the whole point of morphological analysis and systematic "complete field

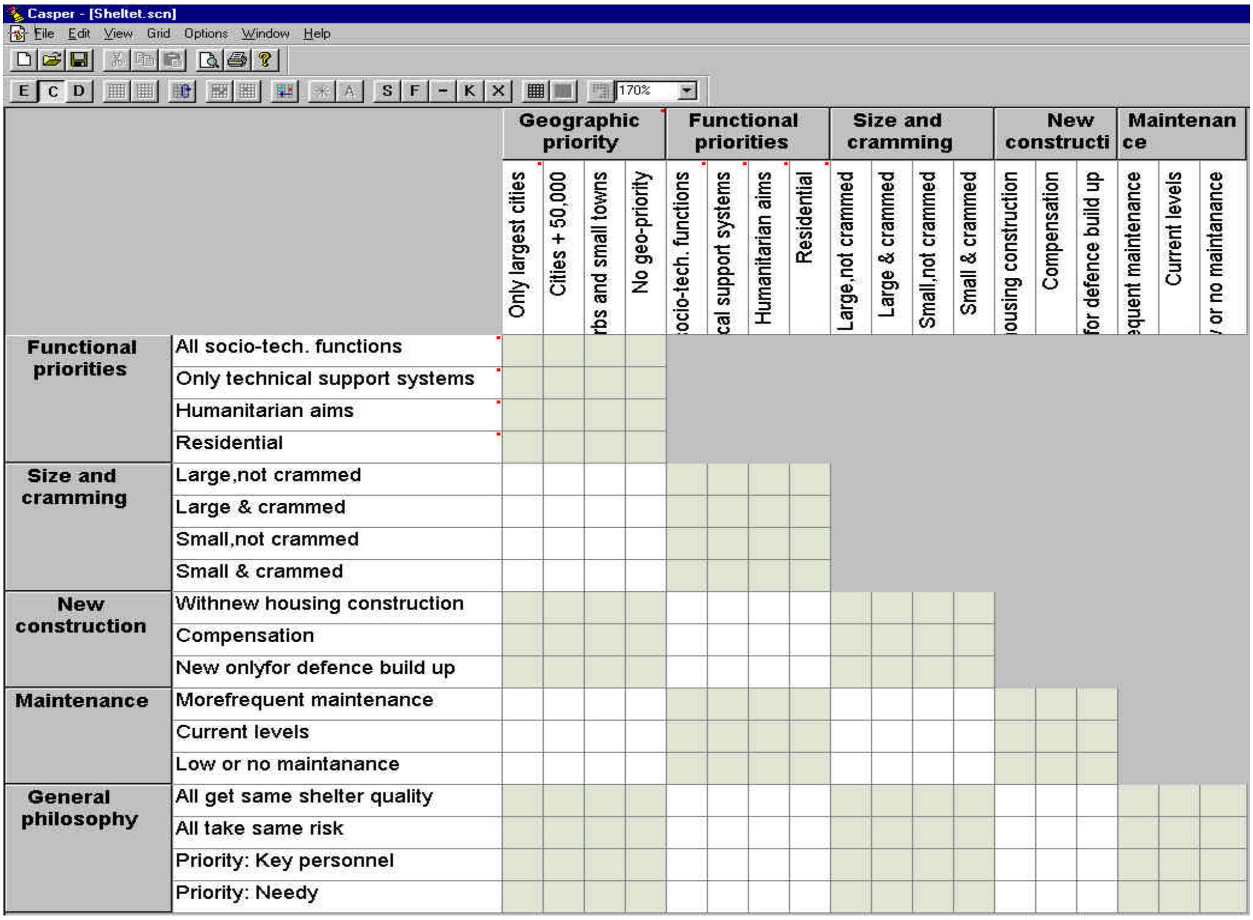


Figure 2: Cross-consistency matrix.

coverage" is to push consciousness to the limits of the conceivable and to facilitate discovery (Zwicky & Wilson, 1967).

The method also has definite advantages for scientific communication and – notably – for group work. As a process, the method demands that parameters, conditions and the issues underlying these be clearly defined. Poorly defined parameters become immediately (and embarrassingly) evident when they are cross-referenced and assessed for internal consistency.

Computer Aided Morphological Analysis

The Swedish Defense Research Agency (FOI) has utilized morphological analysis in some 40 projects during the past 10 years. For this purpose, we have developed software to support the entire analysis-synthesis cycle, which MA involves -- the so-called MA/Casper system.
MA-projects typically involve developing computerized laboratories for generating scenarios and modeling complex systems involving a wide range of disparate, non-quantified variables. Such laboratories have been developed as instruments for *inter alia*: generating threat scenarios and strategy alternatives for the Swedish Armed Forces; identifying alternative long-term social evolutionary trends for the Swedish Nuclear Waste Management agency; evaluating risk and alternative preparedness measures for terrorist actions involving chemical and biological agents.

We have found MA especially suitable in pitting strategy models against scenarios. Thus in such studies, we develop two complementary morphological fields: one for generating different possible scenarios based on factors which cannot be directly controlled (an "external world" field); and one for modeling strategy or system variables which can -- more or less -- be controlled (an "internal world" field).

These two fields can then be linked by cross-consistency assessments in order to establish which strategies would be most effective and flexible for different ranges of scenarios.

Figure 3 is an overlay model, which pits tactical scenarios against a range of artillery systems. From the left, tactical scenarios are expressed as demands placed on artillery systems. From the right, the properties of current and planed artillery systems are expressed.

The marked configuration shows a designated scenario (scenario 3) and its situational demands as "input"; with systems and system-properties as "output".

DAMANDS ON SYSTEMS -----> <----- SYSTEMS PROPERTIES

Casper - [MarkMal.scn]

Tactical scenarios	Purpose	Effect/ penetration	Effect/ precision	Guidance system: final phase	Attaqck attitude	Time to effect after decision to employ	Special weapon system demands/ properties	System configuration
Scenario 1	Destroy	Bunker buster	Great accuracy Little or no side effects	Visual	Vertical	Within 10 sec.	Recognition/ identification capacity	System 1
Scenario 2	Pin down, hinder, stop	Kinetic energy + RSV (Hard)	Great accuracy Limited side effects	IR	Horizontal	Within 1 minute	Command self-destruction (Abort mission)	System 2
Scenario 3	Disrupt	30 mm (medium)	Good accuracy Some side effects	Radar		Within 10 minutes	Updateable target co-ordinates	System 3
Scenario 4	Warn	Small-bore + fragmentation (soft)	Area effect -- 200x300 m	Acustic		Within 30 minutes	Sensor guided warhead	System 4
Scenario 5			Area effect -- 500x400 m	Co-ordinate guided		Within 1 hour	Pre-programmed target co-ords.	System 5
Scenario 6						Within 5 hours	Basic capacity	System 6
Scenario 7						Within 24 hours		System 7
						More than 24 hours		System 8
								System 9
								System 10
								System 11

Figure 3: Two superimposed fields: tactical scenarios vs. system properties.

|<----------------------------- RESOURSE FIELD --------------------------->|<--------- SCENARIO FIELD ---------->|

PLANNING/ PLANS	TRAINING AND EDUCATION	PERSONNEL AVAILABLE	EQUIPMENT AVAILABLE	LEADERSHIP LEVEL	RESPONSE to chemical release	RESPONSE: Information to public	RESPONSE: Affected people
Full preparedness plan	Broad municipal co-operative training	11 or more	Special equipment for specific case	Level 4	Reduce by at least 80% within 15 min	Warn involved within 5 min.	Help many within 30 min.
Response plan for specific case	Training for specific case	8-10	Base equipment for specific case	Level 3	Reduce by at least 80% within 30 min	Warn involved within 30 min.	Help some individuals within 15 min.
Standard routine for specific case	Base education + regular training	5-7	Less than base equipment for specific case	Level 2	Reduce by less than 50% within 15 min	No warning within 30 min.	Help some individuals within 30 min.
Standard routine for general case	Base education only	4 or less		Level 1	Reduce by less than 50% within 30 min		No help within 30 min.
Only alert plan					No measures within 30 min		

Figure 4: Preparedness assessment for accident involving toxic condensed gas (e.g. ammonia).

Figure 4 (above) is an instrument currently being delivered to Sweden's Emergency Rescue Services. It will be used to assess preparedness for chemical accidents and, in its next application, terrorist actions involving the release of chemical agents.

The model is composed of two linked fields: a 6-parameter *resource field* on the left, and a 3-parameter scenario-specific *response field* on the right. The response field can be changed in order to express different accident scenarios and chemical releases. A rescue service "clicks" in its current level of resources (red field) and is presented with an expert-judged response preparedness level (dark blue). The light blue field shows what would be required in the form of improved resources in order to increase response levels. In this case, the resource field is treated as "input", and the response field as "output". The model can also be employed the other way around: a response level can be the designated as input, in order to see what level of preparedness resources would be required.

Conclusions

Morphological analysis, extended by the technique of ”cross-consistency assessment”, is based on the fundamental scientific method of alternating between analysis and synthesis. For this reason, it can be trusted as a useful, conceptual modeling method for investigating problem complexes, which cannot be treated by formal mathematical methods, causal modeling and simulation.

As is the case with all methodologies, the output of a morphological analysis is no better than the quality of its input. However, even here the morphological approach has some advantages. It expressly provides for a good deal of in-built ”garbage detection”, since poorly defined parameters and incomplete ranges of conditions are immediately revealed when one begins the task of cross-consistency assessment. These assessments simply cannot be made until the morphological field is well defined and the working group is in agreement about what these definitions mean. This type of garbage detection is something that strategy analysis and futures studies certainly need more of.

References

Bailey, K. (1994). *Typologies and Taxonomies - An Introduction to Classification Techniques*, Sage University Papers: Sage Publications, Thousand Oaks.

Coyle, R. G., Crawshay, R. and Sutton, L. (1994). Futures Assessments by Field Anomaly Relaxation, *Futures* 26(1), 25-43.

Coyle, R. G., McGlone, G. R. (1995). Projection Scenarios for South-east Asia and the South-west Pacific, *Futures* 27(1), 65-79.

Eriksson, T. & Ritchey, T. (2002). Scenario Development and Force Requirements using Morphological Analysis, *Presented at The Winchester International OR Conference*, England .

Greenstein J. & Wilson A. (1974). Remembering Zwicky. *Engineering and Science 37:15*-19.

Jenkins, L. (1997). Selecting a Varity of Futures for Scenario Development, *Technological Forecasting and Social Change* 55, 15-20.

Rhyne, R. (1981). Whole-Pattern Futures Projection, Using Field Anomaly Relaxation, *Technological Forecasting and Social Change* 19, 331-360.

Rhyne, R. (1995). Field Anomaly Relaxation – The Arts of Usage, *Futures* 27 (6), 657-674.

Ritchey, T. (1991). Analysis and Synthesis - On Scientific Method based on a Study by Bernhard Riemann. *Systems Research 8*(4), 21-41. (Download at: www.foi.se/ma or www.swemorph.com).

Ritchey, T. (1997). Scenario Development and Risk Management using Morphological Field Analysis. *Proceedings of the 5th European Conference on Information Systems (Cork: Cork Publishing Company) Vol. 3*:1053-1059.

Ritchey, T. (2003). Nuclear Facilities and Sabotage: Using Morphological Analysis as a Scenario and Strategy Development Laboratory, *Proceedings of the 45th Annual Meeting of the Institute of Nuclear Materials Management*, Phoenix. (Download at: www.foi.se/ma or www.swemorph.com)

Stenström, M. & Ritchey, T. (1999). Using Morphological Analysis to Evaluate Preparedness for Accidents Involving Hazardous Materials, FOA report. (Download at: www.foi.se/ma or www.swemorph.com)

Zwicky, F. (1969). *Discovery, Invention, Research - Through the Morphological Approach*, Toronto: The Macmillan Company.

Zwicky, F. & Wilson A. (eds.)(1967). *New Methods of Thought and Procedure: Contributions to the Symposium on Methodologies*. Berlin: Springer

Human Behavior Representation

An Algorithm for Learning Tactical Behavior from Observation

Brian S. Stensrud and Avelino J. Gonzalez

Intelligent Systems Laboratory
University of Central Florida
{brian, gonzalez}@isl.ucf.edu

Abstract

This paper describes the components and mechanics of an algorithm that learns context-transition logic for a Context-Based Reasoning [1] model of some tactical behavior. The purpose of the algorithm is to automatically generate a knowledge-base for context-transitions given a sequence of observations of an expert performing a tactical behavior requiring 'high-level' decision-making steps. The knowledge that will be extracted with this algorithm will be 'high-level' in that it will identify situations where the expert has chosen to alter his lower-level course of action or behavior. The situations and goals requiring these lower-level behaviors are considered to be contexts that, when combined along with the overall goals of the expert's behavior (his mission) and the logic that defines when context transitions are necessary (the context-transition logic), represent the complete behavior known by the expert and desired by an autonomous agent tasked to perform that behavior. Assuming that these lower-level behaviors are known and that their corresponding contexts have been encoded, the algorithm proposed will complete the knowledge necessary for an autonomous CxBR model of the behavior, provided that the behavior is well-defined. For this research, a well-defined behavior (or mission) is achieved with a clear description of goals and a logical partitioning of the behavior into contexts. Because of this, the mission goals for the CxBR model must be defined so that they are consistent with the goals assumed by the expert to be observed. Conversely, the context-partitioning must be performed by a knowledge-engineer without any correspondence or interaction with the expert, as the only connection required between the paradigm and the expert's behavior is that the low-level behaviors coded into contexts by the knowledge-engineer be similar in execution to those employed by the expert. The more similar these low-level behaviors can be, the more closely-tied the CxBR model's performance will be to the learning algorithm's ability to generate good context-transition logic. The expert must also be clear on the behavior he is to perform, and must also be in an environment in which he can operate normally and without hesitation. Finally, the observational system must be situated so it has the most direct access to the stimuli on the expert without impeding him. Testing the algorithm requires that all learning takes place within a simulation so that the observing system can have identical perspective of the scenario as the expert.

Introduction

In this section, the components and mechanics of the learning algorithm developed for this research will be explained. The purpose for this algorithm is to automatically generate a knowledge-base for context-transitions given a sequence of observations of an expert performing a tactical behavior. The knowledge that will be extracted with this algorithm will be high-level in that it will identify situations where the expert has chosen to alter his lower-level course of action or behavior. The situations and goals requiring these lower-level behaviors are considered to be contexts that, when combined along with the overall goals of the expert's behavior (his mission) and the logic that defines when context transitions are necessary (the context-transition logic), represent the complete behavior known by the expert and desired by an autonomous agent tasked to perform that behavior. Assuming that these lower-level behaviors are known and that their corresponding contexts have been encoded, the algorithm proposed will complete the knowledge necessary for an autonomous CxBR model of the behavior, provided that the behavior is well-defined.

For this research, a well-defined behavior (or mission) is achieved with a clear description of goals and a logical partitioning of the behavior into contexts. Obviously, the mission goals for the CxBR model must be defined so that they are consistent with the goals assumed by the expert to be observed. Conversely, the context-partitioning must be performed by a knowledge-engineer without any correspondence or interaction with the expert. This is a fundamental design requirement of this algorithm; this research intends to learn from experts that may be either incapable or unwilling to cooperate with the learning task. Furthermore, the expert is not expected or assumed to know anything about contexts, nor is it expected or assumed that the expert reasons in a manner consistent with Context-Based Reasoning. The only connection required between the paradigm and the expert's behavior is that the low-level behaviors coded into contexts by the knowledge-engineer be similar in execution to those employed by the expert. The more similar these low-level behaviors can be, the more closely-tied the CxBR model's performance will be to the learning algorithm's ability to generate good context-transition logic.

Before the learning process can begin, the expert must be clear on the behavior he is to perform, and must also be in an environment in which he can operate normally and without hesitation. Furthermore, the observational system must be situated so it has the most direct access to the stimuli on the expert without impeding him. To test the learning algorithm, all learning takes place within a simulation so that the observing system can have identical perspective of the scenario as the expert.

Context-Based Reasoning

CxBR is a reasoning paradigm that allows an agent to execute a specific, defined task in either a simulated or real-world environment. The task assigned to the agent is encapsulated within a CxBR mission. The mission provides for the agent both a set of goals, which represent the criterion for completing the task, and a set of constraints specific to that task. Also present within a mission is a list of contexts that serve to partition the agent's task-related knowledge.

A context is a situation, based on environmental conditions and agent stimuli, which induces a certain agent behavior specific to that mission. When an agent is executing a mission within CxBR, its behavior is primarily controlled by the current applicable context, (known as the active context) a determination made by context-transition logic. At each time step, this transition logic examines the current stimuli on the agent and makes a determination of the active context for the subsequent time step. This logic is often in the form of sentinel rules that contain the conditions for a specific context-context transition; however the transition logic is not required to be rule-based.

The assignment of a certain CxBR behavior onto an agent (or Autonomous Intelligent Platform, AIP) is done so through the creation of a CxBR model. Encoded within a CxBR model are the missions that are to be assigned to the agent, along with each context applicable to those missions and applicable transition logic. Also present within the model is an interface that connects the knowledge of the CxBR model to the physical or simulated agent executing the behavior. Because of this, the interface used by the CxBR model is representative of the agent it connects to. Low-level functions (such as moving, scanning, etc) are all encoded within this interface along with representations as is the data representing the agent's current state (position, velocity, eye angle, etc). When the model determines a course of action, therefore, it does so in terms of the functions and representations within this interface.

A model is executed by assigning a mission to the agent interface. The first context listed by the mission is denoted the default context, and serves as both the initial context and the context used if no situational determination can be made by the transition logic of the model. When executed, the initial context is either selected by the transition logic or assigned by the mission, and the behavior defined by the model begins. The model executes the logic encapsulated within the current active context, consults the transition logic for the appropriate next context (if a transition is necessary), and repeats until the goal criterion is reached. A description of Context-Based Reasoning is provided in [1].

Acquiring an Observation Sequence

When an expert begins performing a behavior within this simulation, the learning system records all relevant and visible stimuli on the expert along with the actions taken by the expert at the time those stimuli are presented. A recording is taken at each time-step t reached during the execution of the behavior to be learned. To account for the reactive nature of the expert's actions at any time-step t, we will refer to the time at which the stimuli are presented as time t_{0-}, and the time at which the expert switches his active context and chooses a course of action as time t_{0+}. At the point when the expert completes the scenario, the learning system will have compiled a set of recordings that should encompass all relevant stimuli and expert actions taken. This set is known as the observation sequence for the executed scenario. Individual members of this sequence are distinguished by the simulation-time at which they were recorded, and are referred to as observations. These observations denote this time t_0, along with the set of visible stimuli that existed at t_{0-} and the set of actions taken by the expert at t_{0+}.

$$O_{t_0} = \left\langle \Phi_{t_0^-}, \Omega_{t_0^+} \right\rangle$$

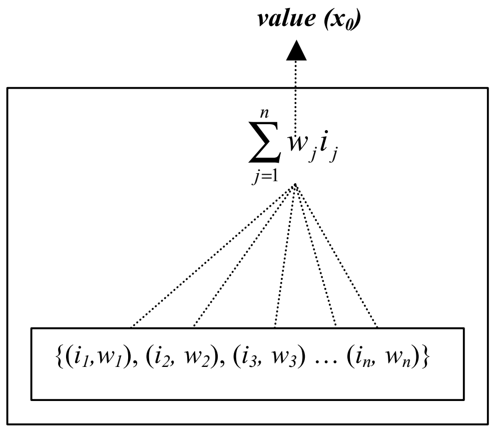
Using TBI to Infer the Current Active Context

The actions and high-level decisions of the observed expert are not tied to the contexts developed to represent the behavior by the knowledge engineer. However, in order to represent the expert's knowledge within a CxBR model, a mapping between the two must exist. To create this relationship, a set of context templates are created by a knowledge engineer prior to generating an observation sequence. Context templates (not to be confused with the templates that exist within Fuzzy ARTMAP) consist of a set of actions and conditions that suggest specific active contexts if they are executed.

Context templates are constructed immediately after a list of contexts for the desired behavior is created. The templates include all potential indicators that the agent executing the behavior is operating within their corresponding context (i.e. that context is active). When the expert is active, a TBI engine will cross-reference his actions with each context template, 'checking off' all indicators that are true. The output of the TBI engine will be the context template whose number and type of indicators 'checked off' by the engine – represented by the value of the template - indicate that it is the most likely candidate for the expert's active context. As discussed earlier, this is an inexact science. For instance, many template indicators that suggest a context being active may not have distinctive true/false thresholds. Furthermore, some indicators might be more important at identifying an active context than others within its context template. For this reason, the templates represented in Gonzalez and Drewes [2] are not appropriate to serve as context templates, and so the context templates first developed in Gerber [3] will instead be used. For his doctoral dissertation, Dr. Gerber used a modified template structure, where templates represented contexts just as in this research, where indicators are represented as fuzzy values between -1.0 and 1.0, allowing for uncertainties amongst each individual active context cue. In addition, he allowed for a weighted sum of each indicator to determine the value of each context template. This was also an improvement over Drewes' notion of a template, where each template's value is computed by a simple sum of indicator 'checks'. With a weighted sum, indicators are able to take on varying degrees of importance depending on the weights they are assigned.

Context Template Structure

Context templates consist of a set of ordered pairs, each pair matching indicator values with their corresponding weight within the template. Indicator values can take on any real value between -1 and 1, depending on the degree at which each indicator exists within the simulation under study. For example, consider an indicator within a context template that denotes the degree to which an agent is facing north. If the agent is facing due north, that indicator would be assigned the value 1. On the other hand, the value -1 might be assigned when the agent is facing due south, or 0 when facing due east or west.



Context Template x_0

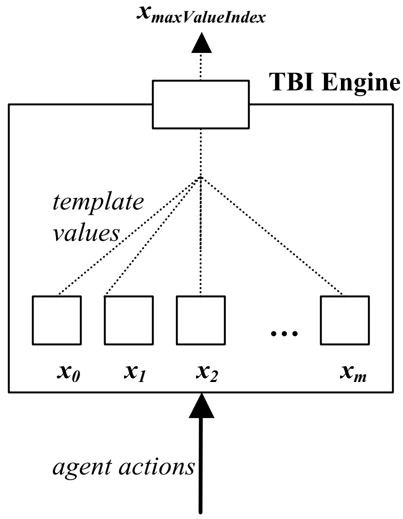


Figure 1 – A Simple TBI Engine

Figure 1 represents a TBI engine with m generic context templates and *n* indicators per template, as developed in Gerber [3] and used for this research. Gerber noted that this structure is an analog to a single-layer backpropagation neural network, which is also the case here save the fact that the weights will be defined a priori by a knowledge engineer. It is important to reiterate here that the crux of this research is to develop an algorithm to learn context transition logic, not to identify contexts. The TBI engine exists in the algorithm to identify active contexts for Fuzzy ARTMAP because it has proven to be successful in the work done by Gerber [3]. If it were possible, a system would be in place that identified precisely the active context at each time step, so that this element of uncertainty could be removed. Here, the scenarios for the expert are such that all contexts can be identified using a TBI engine.

Applying Fuzzy ARTMAP to the Learning Algorithm

The input and output patterns reflect observations recorded at specific times during the scenario along with the active contexts at those times as identified by the TBI engine. Input patterns will be represented by quantitative values for each stimulus on the expert – enemy movements, environmental conditions, current physical conditions, etc. The output patterns will be represented by the action taken by the expert in response to input pattern presented, where each action reflects a transition from the provided context at the input to a new active context (also identified with TBI). The implication here is that every action (and thus every output pattern) will represent a transition to a new context, which is of course not the case. Rather, actions representing no context transition will be represented by patterns that require a transition to the current context – the equivalent to no context change.

A training pattern is generated and presented to a Fuzzy ARTMAP neural network for each observation made of the expert during the execution of a scenario. Fuzzy ARTMAP neural networks are defined and explained in detail in [5]. Learning occurs through the creation of templates in the ART_a and ART_b modules and of a many-to-one mapping between those templates. ART_a templates represent clusters of input patterns, similar in their representation, to which the expert has responded by making a specific context transition. That transition is stored in a template in the ART_b module, and a mapping between the two templates is created. When the network then encounters an input that matches the input pattern cluster represented by that template in ART_a, it will know that the appropriate response is stored in its mapped template in ART_b.

Figure 2 represents the learning mechanism developed for this research. An observation recorded includes both the stimuli on the agent and the actions taken by the expert operator. Each are made at fixed-time intervals at a resolution fine enough to ensure that all decisions made by the expert are recorded and stored as training patterns – in other words, to ensure that no observation or action goes unseen. A decision is considered to be the action made by the expert in response to a set of stimuli presented at time t_0, and is expressed as the context that the agent enters (makes active). This stimuli, along with the active context in which the expert is operating at prior to t_0 (t_{0-}), constitutes the input pattern that is presented to ART_a. The actions that the agent executes in response to these inputs (at time t_{0+}) are analyzed by a TBI module, which then outputs the most likely candidate for the context that corresponds with those actions. That context name is then presented to ART_b as the output pattern for time t_0, and is also stored for time t_1, where it will be presented as part of the input pattern as the active context prior to the stimuli presented and actions taken at time t_1.

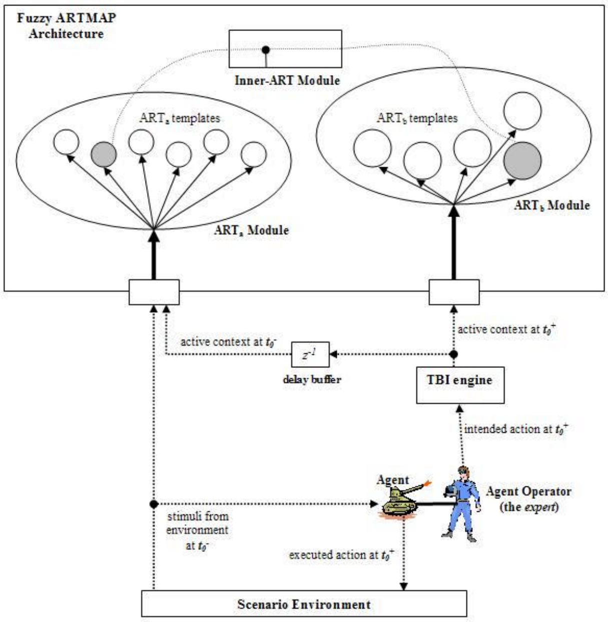


Figure 2 – Learning Context Transitions using Fuzzy ARTMAP

The task for Fuzzy ARTMAP, then, is to learn the correct context transition given the current active context and the input stimuli on the agent. To do this, the architecture will create templates in ART_a that effectively cluster input patterns based not only on their similarities but also on the context transition that the expert makes in response to them. The template corresponding to the actual transition made will be stored in ART_b, and the Inner-ART module will create a link representing a mapping between the two templates. After the training phase is complete, there will exist a many-to-one mapping between the input-pattern templates in ART_a and the context-transition templates in ART_b.

Input Pattern Generation

The input patterns presented to ART_a for time t_0 are constructed from the expert observation recorded at that time along with the active context name inferred by the TBI engine. To be in a usable form for Fuzzy ARTMAP, each piece of this information must be converted to a fuzzy value between 0 and 1.

The specific technique for converting the observation arguments must be determined by the knowledge engineer, and will depend on the nature of the observations required for the learning. To convert the identified active context into a field within the input pattern, one field is set aside for every possible context in the scenario. If a context n is identified as the active context, the n^{th} field is assigned a value of 1, and the other 'context fields' within the input pattern are assigned a 0. This is done to persuade input patterns with different active contexts to bind to different templates in ART_a.

$$\left\{ \overbrace{o_0, o_1, o_2 \ldots o_{n-1}}^{observation}, \underbrace{c_0, c_1, c_2 \ldots c_{n-1}}_{active \cdot context} \right\}$$

Figure 3 - Input Pattern Form for ART_a

Output Pattern Generation

Because the output is simply the context that should be active for the next time-step (the appropriate context transition), the output pattern need only include one field for each possible context as was done for the input pattern. This will make for a trivial clustering task for ART_b, but will prove useful during the testing and execution phase. Representing a context name in this manner allows for the output of ART_b to be both readable and unambiguous to either a knowledge engineer or a separate module created to read its output.

$$\left\{ \underbrace{oc_0, oc_1, oc_2 \ldots oc_{n-1}}_{active \cdot context} \right\}$$

Figure 4 - Output Pattern Form for ART_a

Initial Algorithm Verification

Before using the knowledge acquired by the learning algorithm, it is first necessary to verify that it can correctly anticipate expert reactions while in simulation. To do this, the expert again executes the desired behavior while an observation sequence is recorded. That sequence is converted to a set of input patterns as described above; however the active context at t_{0+} is not presented to Fuzzy ARTMAP at ART_b. The network identifies a template in ART_b to match with the input pattern, and from it (from the mappings created during training) follows the mapping to the corresponding template in ART_b, which contains the information necessary to produce an output. This output will be the name of the appropriate context transition to be made by the expert. Since the neural network is now being evaluated, the actions of the expert (translated by the TBI engine to represent a context transition) that during training were translated into an output pattern are now used as a check of the output generated by Fuzzy ARTMAP. A diagram of this evaluation process is provided as Figure 5.

The evaluation of the neural network is performed by examining how often a match exists between the next active context predicted by Fuzzy ARTMAP and the next active context output by the TBI engine (based on an expert response). Also important to the viability of the learned knowledge is the type of mistakes made by the expert. For instance, consider a scenario where the expert (playing the role of a soldier) is required to hold a position in a bunker; and can only move to assault the enemy, retreat, or move to a different guarded position. In this case, the expert decides to move to a different guarded position but the neural network mistakenly assumes that the correct action (analogous to a context transition for this example) is to assault the enemy. This mistake is likely much more misrepresentative of the expert's behavior than an assumption that the action is to retreat, and should be reflected in the evaluation. This discussion will be continued in a future paper with specific references to the scenario used for this research.

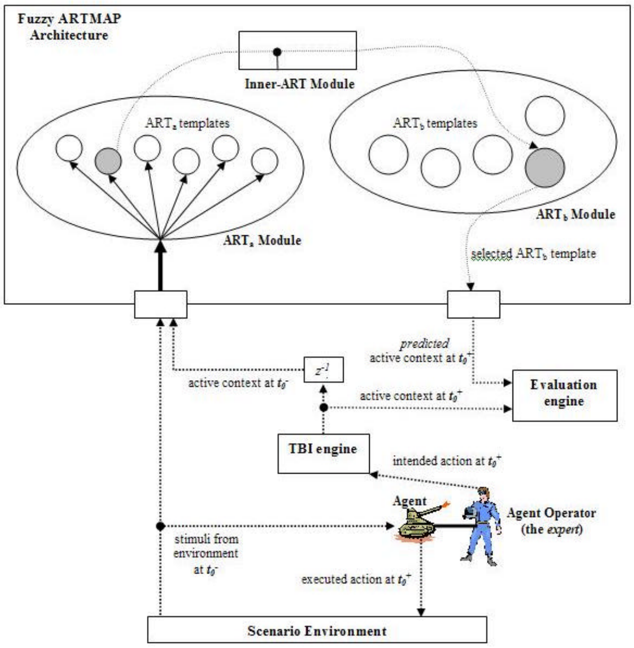


Figure 5 - Evaluating Fuzzy ARTMAP Architecture by Comparing Predicted vs. Actual Context Transitions

Evaluation of the Learning Algorithm

The overall measure of how well the learning algorithm is able to deduce expert decision-making cues must ultimately be determined by allowing an autonomous system to imitate the expert's knowledge in similar scenarios. To do this, the CxBR model – constructed partially by the Fuzzy ARTMAP neural network and partially a priori by a knowledge engineer – is assigned to an AIP that attempts to execute the expert's behavior in a separate physical or simulated environment. If it is assumed that the behaviors represented by the individual contexts closely match that of the expert, the success of the AIP will rely most heavily on the ability of the neural network to generate the correct active context sequence based on the stimuli presented to the AIP during the scenario. It is the degree of this success that this research will use to evaluate the effectiveness of the learning algorithm.

Incorporating Knowledge within a CxBR Model

The second and most important evaluation of the learning algorithm involves the use of the trained Fuzzy ARTMAP network to generate the active context sequence within a CxBR model. The CxBR model, whose individual contexts are written by a knowledge engineer externally to the learning process, in turn controls an AIP to execute the desired behavior performed by the expert in simulation.

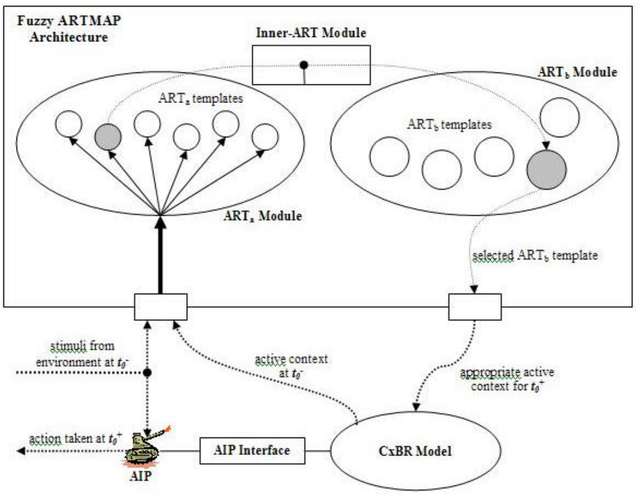


Figure 6 - Determining Context Transitions Using a Trained Fuzzy ARTMAP Neural Network

As illustrated in Figure 6, all that is required to interface the trained neural network into a CxBR model is a consistent naming convention for the active contexts, so that the names can be passed to and from the model and Fuzzy ARTMAP. The CxBR model is then be modified to retrieve the next active context not from a set of transition sentinel rules but instead from the output of the ART$_b$ module of Fuzzy ARTMAP. Aside from the use of the neural network, the CxBR model and the relationship to its controlled AIP remain unchanged.

Results

Initial results of this research are to be included in a future paper, as the testing of this learning algorithm is currently ongoing.

References

[1] Stensrud, B., Barrett, G., Trinh, V. and Gonzalez, A. (2004), "Context-Based Reasoning: A Revised Specification," Submitted to *Florida Artificial Intelligence Research Society (FLAIRS)* Conference, 2004.

[2] Gonzalez, A.J., Drewes, P.J. and Gerber, W.J. (2000), "Interpreting Trainee Intent in Real Time in a Simulation-based Training System", *Transactions of the Society for Computer Simulation International*, Vol. 17, No. 3, September 2000, pp. 135-147.

[3] Gerber, W.J. (2001), "Real-Time Synchronization of Behavioral Models with Human Performance in a Simulation", Doctoral Dissertation, University of Central Florida, Orlando, FL.

[4] Fernlund, H. K. and Gonzalez, A. J. (2001), "An Approach Towards Building Human Behavior Models Automatically by Observation," Intelligent Systems Laboratory, University of Central Florida.

[5] Georgiopoulos, M. and Christodoulou, C. (2001). Applications of Neural Networks in Electromagnetics. Artech House Publishers, 2001.

[6] Henninger, A. (2001). "Neural Network Based Movement Models to Improve the Predictive Utility of Entity State Synchronization Methods for Distributed Simulations", Doctoral Dissertation, University of Central Florida, Orlando, FL.

[7] Van Lent, M. and Laird, J. (1998). "Learning Hierarchical Performance Knowledge by Observation", University of Michigan, Ann Arbor, MI.

Context Driven Reinforcement Learning

David Aihe and Avelino Gonzalez

Department of Electrical and Computer Engineering
University of Central Florida
Orlando, FL

Abstract

Modelling human behaviour is a complicated task that requires the use of knowledge from many domains. It is somewhat universally accepted that this knowledge used in modelling human behaviour has to be extracted from domain experts either through question and answer sessions or through observation. Context-Based Reasoning (CxBR) is a way used in modelling human behaviour. Knowledge from domain experts is represented by the use of Contexts. CxBR is based on the premise that in any given situation, only a few things can realistically happen. Sentinel rules are used in determining the activation and transitioning of contexts. Although this method has been shown to model the behaviour of humans in tactical situations correctly, it has a disadvantage of 'hard coding' the transitioning between contexts based on the knowledge of the domain expert. What can be true and correct in one domain might be totally false and wrong in another domain. In this paper a method for limiting the amount of 'error' or 'conflicting information' introduced by the various domain experts in the contexts is described. The method described here also eliminates the situation where the domain expert has a limited knowledge on the subject being modelled. Reinforcement learning (RL) is what is used to eliminate these problems. RL is a technique used in solving problems that are termed the Reinforcement learning problems. With RL, an agent is allowed to learn by interaction with its environment. The agent receives a reward signal for every goal it does or does not achieve. In this paper it is shown that using RL, the transitioning between contexts would be learned by the agent as it interacts with its environment. There will be no relying on the knowledge introduced by the various domain experts on how to transition between contexts. It will be also shown that the contexts as defined by the experts could be modified during the course of the simulation to reflect the actualities of the environment.

Introduction

Modelling human behaviour is a complex task. It is complex because there are many 'parts' inherent to modelling just one task. So many variables are involved in the way a human makes a decision.

The Oxford dictionary, defines behaviour as 'the actions or reactions of a person or animal in response to external or internal stimuli'. Human behaviours are the actions or reactions of a human in response to some external or internal stimuli. The external stimuli could be but not limited to touch, smell, sight, etc. The way a person reacts to these stimuli dictates his or her actions at that point in time. The representation of how a person would act or react to some stimuli is a task that has confounded researchers for a long time. How do you represent the way a person would act or react in a given situation? This is a question that has no definite answer. Some researchers [Lenox et al., 1999; Kokinov, 1999; Gonzalez et al., 1994; Zachary et al., 1991] have suggested ways to represent how they believe a person should or would react in any given situation. But these representations all have inherent flaws in them given the fact that they are either over simplified by having too many assumptions or they have a partial view or reasoning on how a person should behave in that situation. Also these methods do not address the ever-changing behaviour of humans in a given situation or in a new situation or during a change in situations.

There are many factors that affect the way a human behaves in any given situation. For example you would expect a person being held at gunpoint to cooperate with his or her captors and obey all they say. However could we, in all certainty, represent every person as behaving like that? What about a person who is a martial arts specialist and sees a good opportunity to liberate him/herself from his/her captors, how would this person react? This little example goes to show that there are a countless number of ways a person can behave in a particular situation. The number of variables involved in a person's way of acting is countless and as such trying to address these numbers of variables would lead to unsolvable problems or a representation that doesn't adequately fit the situation. With this in mind, researchers [Gonzalez et al., 1994; Turner and Turner, 1991] have come up with representations to address very specific situations; most of these situations are referred to as tactical situations and have a narrower (limited) number of variables though this limited number of variables could also lead to having a complex problem.

Human Behavioural Representation Through Contexts

Successful attempts have been made by many researchers [Gonzalez and Ahlers, 1994; Turner, 1997; etc] in modelling human behaviours through the use of contexts. Contexts are defined by the Oxford dictionary as "That which surrounds, and gives meaning to, something else". As stated by Sowa, [Sowa, 1999] "the word context has been used with a variety of conflicting meanings in linguistics", Sowa goes on to list two major perceptions of the word contexts as derived from the dictionary. There are:

- The basic meaning of Context is some text that surrounds a word being used in a sentence or some phrase of interest. Sowa suggests this to be a section of linguistic text.
- The derived meaning of Context is in a non-linguistic situation, it includes some topic of interest.

Sowa, goes on to say "Context may refer to the text, to the information contained in the text, to the thing that the information is about or to the possible uses of the text, the information, or the thing itself"[Sowa, 1999]. Due to the many "meanings" of context, some ambiguity results as to what aspect of context should be the central focus. Sowa goes on to suggest some criteria for distinguishing the formal functions of a context based on the informal perception of the context.

Based on the flexibility of contexts, and the many definitions of it, many researchers for e.g. [Dey et al., 1999; Gonzalez and Ahlers, 1995; Turner, 1997; Brézillon, 2002] have proposed methods in which contexts could be used to model intelligent systems, amongst which include human behaviours. Although there is no universally accepted best "model", each model as proposed by the various researchers have their pros and cons. Amongst the various modelling paradigms used to represent human behaviour based on context, is the Context-Based Reasoning paradigm. The Context-Based reasoning paradigm is the focus of this research.

Context-Based Reasoning

Context-Based Reasoning (CxBR) is a technique created by Gonzalez and Ahlers, [Gonzalez and Ahlers, 1994] that provides some intelligence to an otherwise unintelligent agent by controlling its actions in a simulated environment. It is based on some guiding principles that pertain to the way humans think in their everyday activities. The basic ideas from which CxBR was created are [Gonzalez and Ahlers, 1999]:

- In any given situation, tactical experts are proficient at a task by identifying and dealing with only the key features of that situation. An example would be if an automobile is taken to an auto mechanic with water leaking from underneath the radiator, an expert auto mechanic wouldn't bother looking at the tires or the fuel tank to see if there are problems there, because he/she automatically recognises the key feature of the situation, which is water leaking from the radiator.
- The numbers of things that can realistically happen in any given situation are limited. A popular example given by [Gonzalez et al., 1998] is that it is not possible for a tire blow out to occur when a car is waiting in a traffic light and as such an agent wouldn't consider a tire blow out event when in a traffic light scenario.
- When faced with a new situation, the present course of action would be altered accordingly to deal with the present situation. An example would be when driving to work from home; the usual plan of action could be to go from your parking lot to a freeway and then to the parking lot of your office. If a new situation occurs, for example a tire blow out or if the road is blocked due to construction or an accident, a new course of action would be taken to achieve the overall goal of getting to the office.

CxBR is composed of a mission context that defines the overall goals and mission of the agent, one or more major contexts, sub-contexts, sub-sub-contexts, etc.

The mission context defines the objectives and the constraints of the agent. A major context represents actions performed by the agent to achieve its mission. At any point in time, there must be a major context in control of the agent. Major contexts are mutually exclusive of each other. Sub-contexts are used to represent actions not directly critical to reaching the missions objectives; they are usually of short durations and are part of one or more major contexts.

Controlling the agent to achieve its goals is the main objective of CxBR. As the agent performs actions on the environment, the environment changes and a search through the transition rules within contexts is done to recognise each situation. Once a situation is recognised, the context that addresses the situation becomes the active context and takes control of the agent until a change in situation occurs again. This process of situational awareness, action on environment and context transition occurs continuously until the agent achieves its goal or fails to achieve its goal due to no context addressing a particular situation.

One of the drawbacks of the CxBR paradigm is that a detailed knowledge of the environment must be known. Also, the subject matter experts' opinion for any given situation is final. In simple terms, the contexts are 'hard coded' based on the knowledge supplied by the expert and the knowledge of the environment. This leaves little room for flexibility and if the agent gets in a state in the environment that has no context defined by the experts or developers, the agent won't act appropriately. In general, the agents' actions or knowledge are constrained by those of the expert that provides the knowledge for the contexts and the agent doesn't learn.

Attempts have been made to introduce some form of learning to the CxBR paradigm. Gonzalez and Saeki [Gonzalez and Saeki, 2001; Saeki, 2000] introduced the competing context paradigm in which the actions defined in the contexts are not hard coded. The problem with this method is in the time wrap simulation carried out to determine the context to choose.

Fernlund and Gonzalez [Fernlund, 2002] developed an approach that automatically builds contexts by observing human actions. Although a good method, it still has the flaw of depending on the subject matter expert. If the observed expert behaves badly, so will the agent.

Other paradigms exist that use context in modelling human or intelligent agent behaviour. The Context-Mediated Behaviour is another paradigm that was developed by Turner [Turner, 1998]. It is conceptually the same as CxBR with few differences. Instead of searching through the contexts, in Context-Mediated Behaviour (CMB), a diagnosis is carried out on the contexts, also in CMB, contexts are merged to form new contexts, a feature that isn't available in CxBR.

Machine Learning

The oxford dictionary defines learning as "Behavioural modification especially through experience or conditioning." [Dictionary]. [Mitchell, 1997] states that "learning is improving with experience at some task". Dietterich [Dietterich, 2003] states that "machine learning is the study of methods for programming computers to learn". He goes on to argue that although it is relatively easy to develop applications that can be applied to solving a wide variety of tasks, there are generally some tasks in which it is difficult or impossible to do this. He groups these tasks into four categories:

- Problems were no human experts exists
- Problems were human experts exist but cannot explain their expertise because of the implicit nature of what they do
- Problems where the underlying parameters / attributes change rapidly, e.g. the stock market
- Problems that are user specific in a large domain, e.g. a mail filtering system for an organization, each user would have different criteria for filtering junk mails; "self customizing programs" [Mitchell, 1997]

[Dietterich, 2003] believes that machine learning addresses most of the same issues that statisticians, data miners and psychologist address, but the major difference lies on the emphasis placed on the issues. While statisticians want to know and understand how the data has been generated, data miners look for patterns in these sets of data. Psychologists on the other hand try to understand why different people exhibit various learning behaviours and machine learning is concerned mostly about the accuracy and effectiveness of the resulting computer system.

[Dietterich, 2003] states that a learning task can be classified along many dimensions, but believes that an important dimension in which all learning tasks should be classified is the distinction between empirical and analytical learning. Dietterich defines empirical learning as one that relies on some external experience whereas analytical learning requires no external inputs.

Dietterich in [Dietterich, 2003b] draws the relationship between learning and reasoning. He attempts to show the ways in which machine learning research has either incorporated reasoning or left out reasoning.

Machine learning could be divided into three parts; Supervised Learning, Unsupervised Learning and Reinforcement Learning.

Supervised Learning

This is when a teacher is present during training. It is akin to a student going to school everyday and being taught by a teacher. [Christodoulou et al., 2000] states that an essential ingredient of this type of learning is the presence or an external teacher. Usually the learner is given training sets that contain input output pairs. For each input shown to the learning agent, there is a corresponding output given to it. [Dietterich, 2003] notes that a key challenge for this type of learning is generalization. After a few training input – output samples are given to the agent, the learning agent is expected to learn some function that would enable it correctly identify or predict what the output of a new input set would be. Usually the input – output pairs used during training are thought to be independent for proper training to occur.

Unsupervised Learning

This is a scenario where the learning agent is not exposed to an external teacher or critic. The training set is not separated into input – output pairs, hence no target value for any input value. Since no external teacher, [Christodoulou et al., 2000] suggests that a provision be made to identify the quality of the representation that the learning agent is required to learn, and the parameters of the agent are optimized with respect to this measure. After the training is over, the inputs introduced to the agent are grouped based on the similarity measure defined on the learning agent.

Reinforcement Learning

Since the advent of modern computers, it has always been the wish of man to have his computer learn like he does, by learning through mistakes it makes. Reinforcements learning (RL) is primarily learning from experience i.e. learning from your mistakes. RL is also known or referred to as learning by interaction, because you have a reinforcement learning agent that learns what to do at every stage of its decision making by the experiences it has developed from its past interaction with the environment. At first this reinforcement learning agent has no knowledge of what is good or bad and as such acts in a stochastic way in the environment, after receiving positive or negative reinforcements for its actions, it quickly settles on an action that gives it the most reward for any given state. In revisiting the dictionary definition of learning, Reinforcement Learning fits in that definition more than other types of learning.

In RL, a reinforcement-learning agent has to explore new options and also exploit options that it knows to best suit the current situation. One of the issues with reinforcement learning is having the agent balance its choice between exploring new options and exploiting options it knows would give the greatest rewards for the current situation. Sutton and Barto, [Sutton and Barto, 1998] talk about the elements of RL, which include:

- A policy: which defines the way the agent behaves at any time.
- A reward function: which defines what goals the agent has to achieve in the RL problem in an immediate sense
- A value function: defines what goals the agent has to achieve in the long run.
- And sometimes a model of the environment could be an element of the RL problem

The techniques suggested in [Sutton and Barto, 1998] would be summarized here. Sutton and Barto [1998] suggest using dynamic programming to solve the RL problem, it also goes ahead to show ways of achieving an optimal policy and optimal value function, but states the limitation of the dynamic programming method for solving RL problems because it has to have a model of the environment to be able to solve the RL problem.

Another method for solving the RL problem as proposed by [Sutton and Barto, 1998] is the Monte Carlo method. The Monte Carlo method has the features of dynamic programming, with the exception that a model of the environment need not be available.

Another method proposed by Sutton and Barto [1998] for solving the RL problem that was made famous by Tesauro [1995] is the temporal difference learning. The temporal difference learning method is a combination of the dynamic programming and Monte Carlo methods.

Sutton [1999] talks about the current state of RL and also about the feature, stating that researchers should focus on the structures that enable value function estimation. He also states: “how features and other structures can be constructed automatically by machines rather than by people” [Sutton, 1999]. This paper also discusses about the idea of a developing mind as currently being studied in psychology, called constructivism being part of the future. From the work of [Tesauro, 1995], RL was shown to achieve results that other learning methods have not. Tesauro showed that using the TD method as proposed by [Sutton and Barto, 1998] to solve the RL problem, the game of gammon could achieve masters level, this was achieved by the game playing about 1.5 million games against itself and learning from the rewards and mistakes in those 1.5 million games. With this much success of the application of the TD method towards the RL problem with a real life scenario, the RL methods could be said to have great potentials. [Pollack and Blair, 1997] try to say that the reason why [Tesauro, 1995] was successful is because of the domain in which the TD method was used i.e. because of the inherent nature of the game of backgammon. Pollack and Blair carry out another learning method on the game of backgammon and claim that any learning method can achieve what Tesauro did. However they failed to achieve it.

Another Reinforcement Learning technique that seeks to find state action values is Q-Learning developed by Watkins [Watkins, 1989]. In the paper on Q learning by [Cybenko, 1997], an agent in an unknown environment seeks to maximize the total reward it gets when starting from any state by choosing actions that would maximize its total rewards in that state. The problem is that the agent doesn’t know what actions would maximize its total rewards because if it did, the problem being solved would turn out to be a supervised learning problem. Through trial and error, the agent has to learn to choose actions in each state that would bring about a total maximum reward.

Overall most researches in Reinforcement learning are focused on making the learning agent learn some value function or utility of a state or an action faster. As pointed out by [Pratt, 1999] RL may not be a good framework for describing animal intelligence, most works carried out in RL are on grid worlds with the learning agent moving in a north, south, east, west fashion. [Pratt, 1999] also points out the technical hurdles in RL. These hurdles include "Curse of Dimensionality and the slow learning with Primitive Actions." [Pratt, 1999]. In RL, the learning agent doesn't take into account behavioural variances caused by emotions, contexts, etc. In this paper we propose a method that takes into account situational awareness during learning.

Fig. 1 Reinforcement Learning Architecture

Contextual Learning

Most researches on context claim to have some form of learning capabilities. Turner [Turner, 1997] states that the agent used to control the autonomous underwater vehicle (AUV) learns by merging c-schema objects to form new c-schema's that define the current situation. The problem with this is that the agent would always behave the same way under the same conditions even if its behaviour were bad. This in our opinion doesn't amount to true learning. Likewise in the work of Saeki, [2000] true learning doesn't occur because the values provided in the contexts, although soft coded, are usually a range of values provided by experts, which may not be correct. Bonzon [2000] developed a contextual learning model that stores sequences of inference steps that lead to discovery of object-level concepts to be used later. He tries to achieve generality in learning with this approach.

Although Kokinov [1999] didn't explicitly call it learning, his dynamic theory of context in which context is defined as a dynamic state of the human mind can be termed learning. He attempts to show the difference between AI approach to contexts and psychological approach. He states that AI's approach to contexts "may be characterized as navigating between and within the context boxes" [Kokinov, 1999]. He proposes architecture DUAL that implements the dynamic theory of context.

In the authors opinion, true learning of contexts exists when a learning agent is able to adjust and modify its beliefs on its actions in that context, in other words, when a learning agent understands the attributes of the a context and thus modify its actions when in that context through experience, true learning is said to occur.

We propose a method of having a learning agent modify its actions and understanding of any situation by interacting with the environment.

The Method

We postulate a hypothesis here, that given any context (situation with multiple meanings), a learning agent can identify the correct meaning through trial and error interaction with its environment. For example, an agent who is in a driving scenario should be able to learn what 'high speed' means in different contexts, high speed in a 'freeway' context should be different in a 'parking lot' context.

Architecture and Algorithm

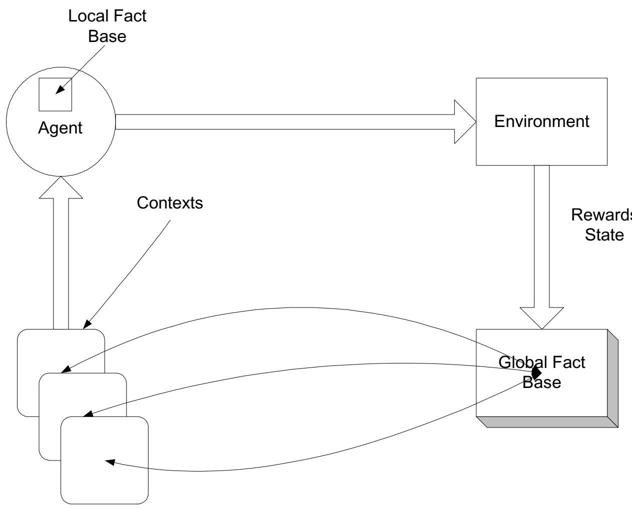


Figure 2 - Context Learning by Interaction (Simplified Version)

Figure 2 shows the simplified version of the Context Driven Reinforcement learning architecture. A more detailed architecture can be seen in the forth-coming dissertation by the author.

The elements in the simplified version are:

- The Learning Agent
- Context library
- Global fact base
- Local fact base

- Environment

The flow of events and actions are as follows:

- The agent performs some action as specified in a default context
- The action performed by the agent changes the state of the environment, from state s_1 to s_2
- The environment returns the current state and the reward associated with that state to the global fact base
- The universal sentinel rules which are always searching for a global change in situation continue firing
- The sentinel rules attached to the active context check the global fact base to see if the defining parameters for the context are still within range
- The global fact base is also checked for the reward available to the agent when that context is active; this reward is stored in the local fact base of the agent, with the context causing the reward and the value of the context that caused the reward.
- If the sentinel rules for the context show that the agent is no longer in that context, the parameters of the compatible contexts / sub contexts are searched, and the sentinel rules for each of the contexts defined as 'next compatible' context or 'sub context' is fired until a matching context is found, this context becomes the active context. (This step is also done by variable substitution of next compatible context and sub-context, the agent is expected to learn what contexts are ideal to transition to)
- This process is carried out until the reward received by the learning agent while in each context is maximized, it is maximized when the optimal value for each variable in the context is reached, i.e., the value of the contextual variable that results in a global maxima for that context.

Algorithm

The algorithm to accomplish the learning from experience in a Context-based structure is as follows

1. A context is activated at random to control the agent.
2. The actions in that context are chosen at random, first on an equal basis, then as the agent knows more about its environment, more weight is given to actions that are known to give a higher reward.
3. There are four actions the agent is capable of doing, increase speed by 10mph, increase speed by 5mph, reduce speed by 5mph and reduce speed by 10mph
4. With a probably of 50%, an action in the active context is chosen
5. Watch the effect of the chosen action on the environment; also get reinforcement from the environment.
6. Store the value of the state, action combination and also the reinforcement received in a look up table in the local fact base of the agent, these values are known as Q values (a neural network can also be used here to enable the agent generalize)
7. For each state of the simulation, choose the maximum Q value
8. When events in the context call for a context switch, freeze the final values of the actions in each state with the reinforcements received and to obtain a context switch follow the same principle, with 50% probability chose a context from the context library to switch to.
9. Repeat these steps until the end of the simulation and until the maximum reward has been achieved in all possible states of the environment.

Experiments and Results

To investigate if our method would work, we carried out an experiment on a learning agent whose mission is to drive from point A to point B.

The environmental coordinates are as follows:

Value / Attribute				
Distance	50	10	40	70
Segment	A	B	C	D
Type	FreeWay	Curve	City	FreeWay
MaxSpeed	70	40	55	70
MinSpeed	40	10	35	40
TrafficLight	0	8	5	0
Activity	N	N	Y	Y

Table 1 Definition of Environment

The meaning of the environment is as follows:

In the first segment, the agent has to go a distance of 50miles, the segment is labelled 'A', if is a freeway, where the maximum speed allowed is 70mph and the minimum speed is 40mph. There are no traffic lights present in this environment and no activities going on.

In the second segment, the learning agent travels a total of 10miles in this segment, thus having the learning agent travel a total of 60 miles from the beginning of its journey. This second segment is a curve with a maximum speed of 40mph and minimum speed of 10mph, there is a traffic light present at 8 miles from the start of the segment and no activities going on.

In the third segment, the learning agent travels a total of 40 miles when in this segment. This segment is a road in the city with a maximum speed limit of 55mph and minimum speed limit of 35mph, there is a traffic light 5 miles into the segment and there are some activities on the road, i.e. car accident or road construction.

Finally in the fourth segment, the agent travels 70miles in the segment and it is a free way segment with some activities on it.

A comparison would be made on how the learning agent would perform using the traditional CxBR and when it learns and modifies the appropriate values in the context. Only one context is considered in this paper to show our method would work, i.e. the free way context.

In defining the context, it is assumed that the expert uses his expert knowledge on previous and similar road segments to define the contexts, i.e. he uses his knowledge of acceptable speed limits, etc from similar road segments. The contexts as defined are as follows:

freeway context

sentinel rules: when a match is found in the global fact base / environment for the road segment type and distance from start of simulation.

subcontext: none

compatible next context: city context, intersection context

actions: increase the speed to max speed limit and maintain max speed limit for the duration of the context

avoids: none

city context

sentinel rules: when a match is found in the global fact base / environment for the road segment type and distance from start of simulation.

subcontext: traffic driving context

compatible next context: freeway context, intersection context

actions: increase or reduce the speed to max speed limit and maintain max speed limit for the duration of the context

avoids: don't exceed max speed limit

intersection context

sentinel rules: when a match is found in the global fact base / environment for the road segment type and distance from start of simulation.

subcontext: traffic driving context

compatible next context: freeway context, city context

actions: increase or reduce the speed to max speed limit and maintain max speed limit for the duration of the context

avoids: don't exceed max speed limit

traffic driving context

sentinel rules: when a match is found in the global fact base / environment for the road segment type and distance from start of simulation.

Subcontext: none

compatible next context: none

actions: reduce speed to zero

avoids: don't run the red light

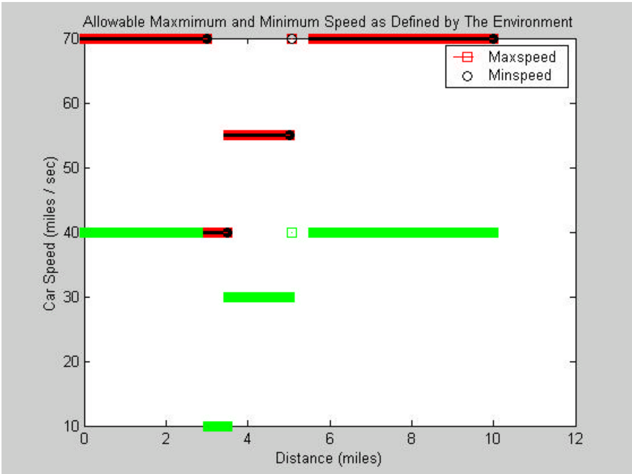


Figure 3 - Allowable Speed Limits as Defined by Environment

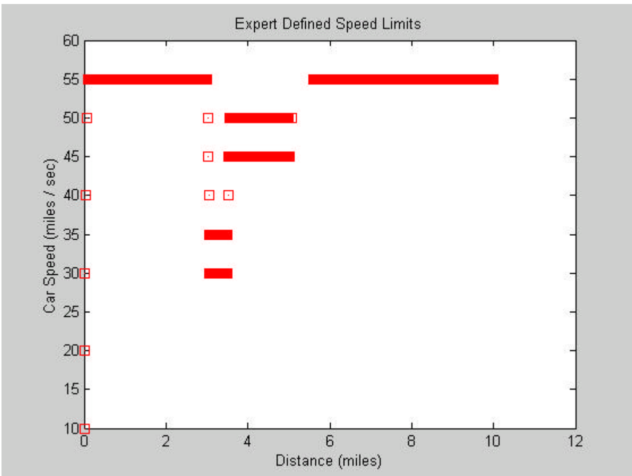


Figure 4 - Expert Defined Speed Limits

As can be seen from figs. 3 and 4, the speed defined in the traditional CxBR is less than the allowable speed for the segment of road. The current knowledge of the domain expert as regards speed limits is different than what the speed limit for the environment really is. The only way for the subject matter expert to know this would have been to review and understand everything about the environment. The only way for the learning agent to achieve knowing what the speed limits are in each segment of road is either having the environment known ahead of time and this 'hard coding' the speed limits or use Reinforcement Learning to have the agent learn what the maximum allowable speed limit is in each segment of the road by trial and error interaction with the environment.

The latter method is better because as soon as any slight change in environment occurs, the learning agent can be retrained and made to learn or unlearn new features of the environment. It also has the added advantage of allowing automatic agent behaviour validation. This is so because different subject matter experts in the same field could have varying opinions as to how to achieve a particular result. If a particular method is used, in order to validate that method, all that needs to be done is to run the simulation a number of times and see how the agent would behave under different scenarios.

Conclusion

As can be seen, the current CxBR methodology doesn't allow flexibility in the way an agent reasons, the contexts would need to be reprogrammed once a slight change in the environment is made, the expert has to be familiar with everything on the environment and also after the CxBR system has been created, other methods would have to be used in its validation.

The method being proposed by the authors show that these shortcomings of the traditional CxBR can be overcome through the use of Reinforcement-Learning. A simplified version of the proposed architecture and algorithm is provided. Preliminary results of the shortcomings of the original CxBR method are also shown in a simple mission of moving from point A to point B.

The next steps in the research include performing actual experiments to validate our architecture and algorithm. Conceptually the algorithm and architecture look good and initial results obtained are encouraging.

References

Bonzon, P. (2000). *Contextual Learning: Towards Using Contexts to Achieve Generality*. Vol. 10, pp. 127-141, Kluwer Academic Publishers.

Brezillon, P. (1999). Context in Problem Solving: A Survey. *The Knowledge Engineering Review, 1*(14), 1-34.

Brezillon, P. (2002). Modeling and Using Context: Past, Present and Future.

Christodoulou, C. & Georgiopoulos, M. (2000). *Applications of Neural Networks in Electromagnetics.* Artech House Inc.

Corkill, D. D., Gallagher, K. Q. & Johnson, P. M. (1987). Achieving Flexibility, Efficiency, and Generality in Blackborad Architectures. *AAAI* 18-23.

Corkill, D.D. (1991). Blackboard Systems. *Journal of AI Expert, 6*(9), 40-47.

Cybenko, G., Gray, R. & Moizumi, K. (1997). *Q-Learning: A Tutorial and Extensions*. Mathematics of Neural Networks, Kluwer Academic Publishers, Boston/London/Dordrecht

Dey, A., Abowd, G. & Salber, D.(1999). A Context-based Infrastructure for Smart Environments. In *Proceedings of the 1st International Workshop on Managing Interactions in Smart Environments* (MANSE '99), pp. 114-128, December 13-14, Dublin, Ireland

Dietterich, T. G. (2003). Machine Learning. *In Nature Encyclopedia of Cognitive Science.* Macmillan, London

Dietterich, T. G. (1997). Machine Learning Research: Four Current Directions. *AI Magazine, 18*(4), 97-136

Dietterich, T. G. (2003). Learning and Reasoning.

Fernlund, H. & Gonzalez, A. (2002). An Approach Towards Building Human Behaviour Models Automatically by Observation. In *Proceedings of the 1st Swedish–American Workshop on Modeling and Simulation* (SAWMAS-2002), Orlando, Florida.

Gonzalez, A. J. (1999). Validation of Human Behavioural Models. In *Proceedings of the 12th International Florida Artificial Intelligence Research Society Conference*, pp. 489-493, May 1-5, Orlando, Florida.

Gonzalez, A. J. & Ahlers, R. (1999). Context-based Representation of Intelligent Behaviour in Training

Simulations. *Transactions of the Society for Computer Simulation International, 15*(4), 153-166

Gonzalez, A. J. & Ahlers, R. (1995). Context-based representation of intelligent behaviour in simulated opponents. In *Proceedings of the 5th Conference on Computer Generated Forces and Behavioural Representation,* pp. 53-62

Gonzalez, A. J. & Ahlers, R. (1994). A Novel Paradigm for Representing Tactical Knowledge in Intelligent simulated Opponents. In *Proceedings of the 7th international conference on Industrial and engineering applications of artificial intelligence and expert systems,* pp. 515-523, Austin, Texas.

Gonzalez, A. J., Georgiopoulos, M., DeMara, R. F., Henninger, A. E. & Gerber, W. (1998). Automating the CGF Model Development and Refinement Process by Observing Expert Behaviour in a Simulation. In *Proceedings of the 7th Conference on Computer Generated Forces and Behaviour Representation*, July, Orlando, Florida.

Gonzalez, A. J. & Saeki, S. (2001). Using Contexts Competition to Model Tactical Human Behavior in a Simulation. *CONTEXT 2001*, pp. 453-456

Gonzalez, F. G., Grejs, P. & Gonzalez, A. J. (2000). Autonomous Automobile Behaviour through Context-Based Reasoning. In *Proceedings of the 12th International Florida Artificial Intelligence Research Society Conference*, pp. 2-6, May 22, Orlando, Florida.

Harmon, M. E. & Harmon, S. (1996). Reinforcement Learning: A tutorial

Kaelbling, L. P., Littman, M. L. & Moore, A. W. (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research, Vol. 4*, 237-285

Keerthi, S. & Ravindran, B. (1995). A Tutorial Survey of Reinforcement Learning. *Indian Academy of Sciences,* Sadhana.

Kokinov, B. (1997). A Dynamic Theory of Implicit Context. *In Proceedings of the 2nd European Conference on Cognitive Science*, April 9-11, Manchester, UK

Kokinov, B. (1999). Dynamics and Automaticity of Context: A Cognitive Modeling Approach. In *Proceedings of Modeling and Using Context, Second International and Interdisciplinary Conference*, CONTEXT'99, pp 200-213, September, Trento, Italy.

Lenox, T., Payne, T., Hahn, S., Lewis, M. & Sycara, K. (1999). MokSAF: How should we support teamwork in human-agent teams? *CMU Technical Report. CMU-RI-TR-99-31*

Mitchell, T. (1997). *Machine Learning*. McGraw-Hill

Pollack, J. B. & Blair, A. D. (1997). Why did TD-Gammon work? *Advances in Neural Information Processing Systems*, Vol. 9, 10-16

Pratt, E. J. (1999). Elephants Don't Play Backgammon Either. *MIT Leg Laboratory, 545 Technology Square, Cambridge, MA 02139*

Russell, S. & Norvig, P. (1995). *Artificial Intelligence: A Modern Approach.* Prentice Hall

Saeki, S., & Gonzalez, A. J. (2000). Soft-coding the Transitions between Contexts in CGF's: The Competing Context Approach. In *Proceedings of the Computer Generate Forces and Behaviour Representation Conference*, Orlando, FL, May 17, 2000.

Sowa, J. (1999). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. New York, PWS Publishing Co

Sutton, R., & Barto, A. G. (1998). *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA.

Sutton, R. (1999). *Reinforcement Learning: Past, Present and Future*. Springer-Verlag, pp. 195-197

Tesauro, G. (1995). Temporal Difference Learning and TD-Gammon. *Communications of the ACM, 38*(3), 58-67

Turner, E. H. & Turner, R. M. (1991). A Schema-based Approach to Cooperative Behaviour. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society.*

Turner, R. M. (1993). Context-Sensitive Reasoning for Autonomous Agents and Cooperative Distributed Problem Solving. In *Proceedings of the IJCAI Workshop on Using Knowledge in its Context,* Chambery, France.

Turner, R. M. (1997). Context-Mediated Behavior for Intelligent Agents. *International Journal of Human-Computer Studies, 3(*48), 307-330

Turner, R. M. (1998). Context-Mediated Behavior for AI Applications. In *Proceedings of the 11th International Conference on Industrial and Engineering Applications of*

Artificial Intelligence and Expert Systems, IEA/AIE-98, Vol. 1, pp. 538-545, June 1-4, Castell, Spain.

Watkins, C. J. C. H. (1989). Learning from Delayed Rewards. Ph.D. thesis, King's College, Oxford.

Zachary, W., Ross, L., & Weiland, M. (1991). COGNET and BATON: An integrated approach for embedded user models in complex systems. In *Proceedings of International Conference on Systems, Man, and Cybernetics,* (Vol. 1, p. 689). New York, NY: IEEE.

Warfare Training With Game Consoles:

Training without Getting Your Hands dirty!

Jenny Lindoff and Håkan Hasewinkel

Swedish Defence Research Agency (FOI)
P. O: Box 1165, SE-581 11 Linköping, Sweden
jenny.lindoff@foi.se and håkan.hasewinkel@foi.se

Abstract

Over the last decade the view on games has changed. Computer games are no longer only used for entertainment, but also for scientific and educational purposes. During the last three years several studies with commercial games have been carried out at the Swedish Defence Research Agency (FOI), at the department of Man-System-Interaction (MSI), to study decision making and different command and control issues. Efforts have also been made to investigate if commercial games can be integrated with existing military systems and used for training and education. In August 2003 a comprehensive study was carried out with over 50 cadets at the Swedish Army Combat School, MSS Kvarn, in Linköping. The aim was to investigate if it is possible to combine a field exercise with a similar exercise performed in a game based environment with COTS products. The game Ghost Recon was used on three different game consoles and on PC. The cadets' mission was to carry out different tasks involving communication, approach march and enemy contact in combat pairs, both in a live fire exercise and in the game based simulation. Both exercises were supervised by instructors from MSS Kvarn. After the exercises the participants filled out a number of questionnaires. Results show that most participants are positive towards the idea of combining field exercises with game based simulations in training of certain tasks in their education. The participants stated that game based simulations can be a useful tool to practice skills like giving orders, skills that they can later use in the field. The results also show that the participants believe it can be useful to practice communication in a calm environment even though they find it problematic not being able to use hand signals and visual signals in the same way they do in the field. The main conclusion from the study is that games like Ghost Recon, with no modifications, with advantage can be used for combat pair training. However, the role of the instructor is very important. An interesting future work would be to conduct a similar study with squad leaders.

Introduction

Many people find different kinds of games amusing and entertaining – something they do in their spare time. In Sweden students use computers more and more, and they spend, on average, more than two hours a day on entertainment[1]; playing games, listening to music, editing digital photos, ect. (see SIKA; SCB). However, computers and computer games are not solely used for entertaining purposes. Over the last decade the view on games has changed. Computer games are no longer only used for entertainment, but also for scientific and educational purposes (Calvert, Jordan & Cocking, 2002; Crawford, 2003; Macedonia 2001). During the last three years several studies with commercial games have been carried out at the Swedish Defence Research Agency (FOI), at the department of Man-System-Interaction (MSI), to study

[1] The average time (127 minutes) is based on male students, in the age 16-24, that have a computer

decision making and different command and control issues. A number of different games have been used for different purposes. Among other things, several first person shooter (FPS) games have been used. In the spring of 2001 a study with Delta Force 2 was carried out (Kylesten & Hasewinkel, 2001). The purpose of that study was to evaluate these kinds of games and to investigate whether or not they can be used to study issues like command and control, decision-making, cooperation and communication. The results from the study were encouraging and showed that FPS-games have many good qualities and can, in many respects, be view as a micro world or a low budget simulator, as long as their limitations and shortcomings are taken into account (Kylesten & Hasewinkel, 2001). Since then, FPS-games have been used for several different studies in Sweden at FOI. During the summer of 2001 a comprehensive team performance study with 120 participants was conducted with the game Operation Flashpoint (Berggren & Andersson, 2003). The purpose of the study was to investigate how mutual understanding within a team, i.e. mutal expectations on performance affects team performance. In order to study this phenomenon a dynamic environment was needed where the participants were forced to take action continuously throughout the experiment. Operation Flashpoint offered such an environment.

Efforts have also been made to investigate if commercial games can be integrated with existing military systems and used for training and education. One of the goals is to investigate whether or not it is possible to use the traditional command and control systems on commercial game hardware, for example Xbox, to reduce the need for expensive computers. This was initially tested in the spring of 2003 at K4, Norrlands Dragoon Regiment, in Arvidsjaur in the north of Sweden. In this study three separate taskforces were led by a ranger staff battalion who operated their ordinary command and control system. The taskforces had three separate missions that they carried out in three different game based simulations (GBS) using three different FPS-games; Delta Force, Ghost Recon and Rough Spear. Each taskforce were connected to a LAN of either Xboxes or PCs. The exercise turned out well and the participants were positive towards the idea of integrating game technology with ordinary exercises (Wikberg, Hasewinkel, Lindoff, Stjernberger & Ericsson, 2003).

It can be concluded that games have many good qualities and in some respects they can be viewed as low budget simulators. However, little research has been conducted on *combining* traditional training with game based training and also investigating what educational effects can be gained.

A Study with FPS-games

In August 2003 a comprehensive study was carried out at the Swedish Army Combat School, MSS Kvarn, in Linköping. The purpose of the study was to investigate if it is possible to combine a field exercise with a similar exercise performed in a game based environment with COTS[2] products. Primarily three aspects were to be evaluated in the two different environments; *performance*, *similarity* and *educational effect*, i.e. we wanted to see how the participants performed, to what extent they could use the same skills in both environments and also investigate whether or not they found the exercises educative. In the study 55 cadets participated[3] (53 men and 2 women). Their average age was 23,13 years (std: 2,06, min: 20, max: 31). The cadets' mission was to carry out different tasks in combat pairs, i.e. they were to practise communication, approach march and enemy contact, both in a live fire exercise and in a game based simulation, GBS. A within-group design was used, i.e. all cadets participated in both the field exercise and in the GBS. Sixteen cadets participated in the GBS first and 39 cadets started with the field exercise. Each exercise, including time to fill out a number of questionnaires, took approximately four hours, thus all combat pairs participated roughly eight hours.

The live fire exercise was part of the cadets' ordinary education program and was planed and led by instructors from MSS Kvarn. During the exercise the cadets were to practice skills like communication, approach march and enemy contact. The cadets worked in combat pairs and participated in groups of approximately 20 people.

The GBS was initiated by researchers from the Swedish Defence Research Agency, FOI. To make sure that the GBS would become meaningful and somewhat realistic it was organized and carried out in cooperation with MSS Kvarn. Tom Clancy's game *Ghost Recon* (see Ubi soft entertainment, 2003) was used unmodified on three different consoles (Playstation 2, Xbox, Game Cube) and on PC[4]. The game was chosen because it is available for all four platforms. Also, it takes place in an environment that is fairly similar to the environment the participants encounter in the live fire exercise and it is possible to apply some tasks, or training objectives, in both environments.

Before the GBS was carried out the cadets were asked to fill out a short questionnaire. This was done to assess their

[2] Commercials Of The Shelf, i.e. products that are available to consumers, usually to a fairly low price.

[3] A total of 56 cadets participated in the game based simulation, but only 55 participated in the field exercise. Since the study is based on the participants' experiences of *both* exercises, the results from the 56[th] participant was excluded.

[4] The study had a second aim; to evaluate pros and cons between different game consoles, which is why four different platforms were used. The results from that evaluation will not be discussed in this paper.

computer and gaming experience. In the GBS the cadets worked in combat pairs and participated in groups of eight people. Each combat pair had to solve three objectives on each platform within a time limit of 20 minutes per platform. Before each mission was carried out the cadets got to practice in a training scenario. The conditions for the mission, and orders, were given by MSS instructors. The instructor also emphasized that the purpose of the exercise was not to "win the game" but to handle the mission in accordance with military rules and regulations, i.e. they where instructed to use correct commands when they communicated and to advance in an appropriate way. Three different instructors participated in the study, thus, the same instructor was not present during all GBS sessions.

It should be mentioned that the GBS was not part of the cadets' ordinary education program and therefore the cadets' did not get as much guidance and feedback as they would normally get in a training session. The primary purpose of the study was to explore the possibilities of using COTS products for training exercises and to investigate whether or not it is suitable to combine field exercises and GBS'. Thus, the live fire exercise was more focused on training than the GBS.

When the cadets had carried out both the live fire exercise and the game based simulation they filled out a questionnaire. In the questionnaire they rated their own communication, approach march, enemy contact and their over all performance in both exercises. They also rated to what extent they found the exercises educative, meaningful and stressful. Thus, the results in this study are based on the cadets' subjective ratings and experiences.

Results and Discussion

The results show that the cadets spend on average 8.12 hours per week in front of the computer (std. 7.26, min: 1, max: 30). They spend, on average, 3.90 hours per week on gaming, i.e. playing PC-games and consol-games, (std. 5.58, min: 0, max: 29). Thus, the computer and gaming experience among the cadets is fairly low, and largely spread, compared to the average male student who spend approximately 14 hours a week (see SIKA; SCB). However, the cadets represent a homogenous group that is not fully comparable to the average male student.

Results also show that the amount of hours the participants spend in front of the computer correlates positively with their performance in the GBS: .293*. i.e. the more time the participant spend in front of the computer the better he/she performs. The time they spend on gaming also correlates positively with how they perform in the GBS: .398**, i.e. the more they play games the better they perform.

In the study the participants rated their own overall performance both in the GBS and in the live fire exercise. The results show that the participants feel they performed better in the field exercise than in the GBS. Their average rating[5] was 5.61 in the field (std: 0.81) and 4.78 in the game based simulation (std: 1.23). Mean comparison revealed that the difference was significant, $t(47)=3.48$, $p<0.05$. Some cadets stated that the live fire exercise was fairly basic and suitable for their educational level. Several cadets mentioned that the GBS was difficult due to limited gaming experience.

Orientation and interaction difficulties. Many participants stated that it was difficult to orientate and to keep track of their combat pair partner in the game based environment. They thought it was much easier to work together and co-ordinate fire and movement in the field exercise. Many participants also mentioned that it was difficult to handle the handheld devices for the different platforms (i.e. gamepads, keyboard and mouse) in a good way due to inexperience.

There are a number of ways to overcome the interaction difficulties in the GBS. Part of the problem would probably disappear just by letting the participants practice more. Further more; in this study *four* different platforms were used, i.e. the participants had to learn how to use four different interaction devices that had basically the same hardware structure but different kepads with different functions. Thus, if one platform is being used instead of four, the time spent on practise can probably be reduced.

The problems relating to orientation and keeping track of the combat pair partner is probably more dependent on your choice of game than of how much time you spend on practice. The game Ghost Recon is a FPS-game where the player has a pretty limited field of vision, i.e. the player has practically no peripheral vision. Therefore it is very difficult to keep track of a team member or to discover an enemy. The field of vision is different in different games so you can minimise this problem by choosing another game. However, even if you choose a different game, one important orientation related problem still remains. When you "look around" in the simulated world it is easy to loose orientation because the participant does not move. Movement in the game is controlled with a gamepad, by means of a game compass. This is a well known, classical problem, which has been studied by several research labs (MOVES; ICT and FOI).

* Correlation is significant at the 0.05 level (2-tailed).

** Correlation is significant at the 0.01 level (2-tailed).

[5] In all ratings a scale from 1 to 7 was used, where 1 was the lowest rating (very bad, totally different, not at all educative ect.) and 7 was the highest rating (very well, exactly the same, very educative ect.).

Task performance. In the exercises the participants had a number of training objectives, i.e. they were to practise communication, approach march and enemy contact. Results show that the cadets rated their task performance higher in the field than in the game based simulation, see table 1.

Table 1 Task performance. The table shows the participants' ratings on their task performance in the field and in the GBS.

	Field	GBS	
Communication	5,64	5,32	5,48
Approach March	5,7	4,52	5,11
Enemy Contact	5,5	4,5	5,00
	5,61	4,78	

Task performance was analysed by an ANOVA 2x3 within-participant design. The main effect of the exercise environment, i.e. field vs. GBS, revealed that the difference was significant $F(1.98)=19.18$, $p<0.05$, $MSe=0.61$, thus the participants rated their performance higher in the field than in the GBS. The second main effect, i.e. type of task, $F(2.98)=10.44$, $p<0.05$, $MSe=0.61$, revealed that the different tasks were rated differently. Pair-wise comparisons show that the difference between approach march and communication is significant, mean difference=.370, $p<0.05$, and the difference between communication and enemy contact is significant[6], mean difference=.480, $p<0.05$. In other words, the cadets believed they performed best with their communication and had more difficulties with their approach march and enemy contact. The interaction effect revealed, $F(2.98)=9.16$, $p<0.05$, $MSe=0.56$, that the cadets rated approach march and enemy contact lower, and communication higher, in the GBS.

One possible explanation of the lower rating in the GBS is the above discussed lack of experience and interaction difficulties. The lower rating was evident on approach march and enemy contact, *not* on communication. This is not very strange since communication is not effected by gaming experience in the same way as the other two tasks, i.e. even though a participant have difficulties related to the interaction he/she can still be good at giving orders and communicating with his/her combat pair partner. Another possible explanation is that the participants not only judged their task performance in the GBS on how well they handled the training objectives but also took into account if they played well, i.e. if they "won the game", which was irrelevant and difficult to achieve within the given time limit. This "winning condition" is in the nature of games and something that you normally do not find in traditional simulation platforms.

[6] the comparisons are tested under the LSD criterion criterion

Many cadets stated that it was difficult to handle enemy contact both in the field exercise and in the GBS. In the field they found it difficult to use hand grenades and to give correct target indication. In the GBS the cadets found it difficult to discover the enemy and judge its location and distance. They also found it difficult to coordinate their fire since they did not always know exactly where their combat pair partner was.

Similarity

In the questionnaire the participants also answered questions regarding similarity, i.e. the participants were asked whether or not they handled the tasks in the same way during the field exercise and the game based simulation. The results show that the participants do not think they carried out their tasks in the same way during both exercises[7], see table 2.

Table 2 Similarity. The table shows to what extent the participants think they handled their tasks in the same way during both exercises.

	Mean	Std. Deviation
Communication	3,3	1,3
Approach March	4,69	1,44
Enemy Contact	3,98	1,6

The results also show that the participants think that the approach march was most similar during both exercises whereas the communication was least similar. Paired sample t-tests show that the differences were significant, $t(51)=3.55$, $p<0.05$, $t(49)=2.64$, $p<0.05$, respectively. What makes these results interesting is the fact that the participants rated their *communication performance* highest (see table 1) but *communication similarity* lowest, i.e. they thought they communicated well during both exercises but they did not communicate in the same way during the field exercise and the GBS. One important question that needs to be raised is if the cadets really practice the same communication skills in the field exercise and in the GBS? Many participants stated that they found it difficult not being able to use hand signals and visual signals in the GBS in the same way as in the field, thus they could not communicate in the same way during both exercises. On the other hand, many participants thought it was good to be able to practice target designation and giving orders in a calm environment, thus they found both exercises useful even though they could not communicate in exactly the same way. The instructors that were present during the GBS

[7] According to the scale used for this question 7 represented "exactly the same" and 1 represented "not at all" so if the rating ? 7 the participants did not handle the tasks in the same way during both exercises.

noticed that the cadets' communication skills got better and better over time which indicates that the GBS gave some kind of training effect. However, it is still important to investigate what kind of communication skills the cadets actually practise in the GBS and if those skills can be applied in the field.

Educational effect

The cadets also rated to what extent they found the exercises educative. The results show that the participants believe that the field exercise was more educative than the GBS, mean value 5.64 (std: 1.24) vs. mean value 3.69 (std: 1.43). Paired sample t-test shows that the difference was significant, $t(54)=8.69$, $p<0.05$. One possible explanation for this result is that the primary purpose of the GBS was not to be educative and the participants did not get as much guidance and feedback as they got in the field exercise. Some cadets stated that the purpose of the GBS was not clear and therefore they did not find the exercise very educative. Some cadets also criticised the game and stated that the exercise might have been more educative if a different game had been used. This is something we are aware of and agree with.

Another possible explanation concerns the participants own ratings on their performance. Since they rated their performance lower in the GBS than in the field they might have found it less educative. To investigate whether or not this is true correlation tests were conducted. The results show that the participants ratings of how educative the GBS was does in fact correlate with how they assessed their own performance: .460**, thus, the participants that felt that they did not perform very well in the GBS (possibly due to inexperience) did not find the exercise very educative.

Many participants stated that field exercises, in general, are *invaluable* because during field exercises they get to practice in the right environment, with the right equipment. Some cadets stated that "learning by doing" is the best way of learning and therefore field exercises are better than GBS'. The cadets do, of course, have a point. We *do not* propose that a GBS is better than a field exercise and we *do not* propose that the GBS should replace field exercises. What we do want to investigate is if it is good to *combine* GBS' and field exercises. Maybe it is good to practice some exercises in a GBS and then go out into the field, i.e. to use the GBS as a way of preparing the cadets for the field exercise so they learn as much as possible.

The cadets' attitude towards the GBS' was not all negative; they did find them educational for some tasks. They stated that they believe that a GBS is a good environment to practice tactics, cooperation, decision making and giving orders.

Combining exercises. So, is it good to combine field exercises and game based simulations? Results show that the participants think that it is good to combine field exercises with GBS[8], mean value 4.47 (std: 1.75). One-sample t-test shows that the result was significant, $t(54)=10,70$, $p<0.05$. Even though the average rating is just slightly higher than four, the result should still be considered to be positive. Since the GBS was not designed to give a training effect or to be educative it is reasonable to assume that the participants would have been even more positive if they would have gotten more guidance and feedback - there are in fact results that support this assumption.

The instructor is important. During the GBS we observed that the participating instructors gave a different amount of feedback to the cadets, i.e. one instructor gave considerably more guidance and feedback than the other two instructors. To investigate whether or not the amount of guidance and feedback influence the participants' attitude towards the idea of combining field exercises and simulations, mean values between the different instructor groups were compared. The results show that the group that got a lot of feedback (eight cadets) think that the game based simulation was *more educative* than the rest of the cadets, mean value 4.25 (std: 1.49) vs. mean value 3.59 (std: 1.42). Further more, they were more positive towards the idea of *combining* field exercises and GBS, mean value 5.63 (std: 0.52) vs. mean value 4.28 (std: 1.83). The ANOVA revealed that the difference between the participants' attitude towards combining exercises was significant, $F(1.52)=4.17$, $p<0.05$, $MSe=2.95$. The difference in rating on how educative the exercise was was not significant.

Conclusions

Even though the study had limitations relating to the game at hand, the interaction problems that occurred and the limited amount of feed-back the cadets received, a number of valuable lessons have been learned regarding the use of FPS-games in training of Swedish army cadets.

Firstly, although no modifications were made in the game, the study was successful and appreciated by the cadets and instructors. Considerable improvements could have been gained if the game would have had an open- or semi open source code making it possible to develop extra or own functions. Done correctly an inexpensive pedagogical and statistical tool could be obtained that can be used for training and education.

Secondly, the instructor's role, and attitude, is important because it affects the GBS and the participants. The instructor must plan the GBS as any other military exercise

[8] According to the scale used for this question 4 represents "not good nor bad", thus a value higher than 4 can be considered to be more good than bad.

so that it is considered to be more than just a game despite the game based interface.

Thirdly, the GBS is a good compliment to field exercises. The GBS offers a good environment for combat pairs to practise basic skills like communication and giving orders. However, it ought to be even more interesting to use the GBS for squad leadership training.

Future work

In the GBS it is easy to create complex, dynamic situations where the squad leader can practice various skills. Squad leader training is highly relevant for Swedish units participating in, for example peace keeping operations in urban environments. Therefore, it would be interesting to use Full Spectrum Warrior as a game based training tool for leadership training of squad leaders, a study that could be designed as a comparative study involving Swedish- and U. S. Army cadets, - ending in a paper presentable at SAMWAS 06.

References

Berggren, P. & Andersson, J. (2003) Team performance: the relationship between shared mental model measures and team performance. Presented at *Human factors of decision making in complex systems.* Dunblane 8-11 September, 2003.

Calvert, S., Jordan, A. & Cocking, R. (2002) *Children in the digital age.* Praeger Publisher, Westport, USA

Crawford, C. (2003). *On game design*. New Riders Publishing Indiapolis, USA

ICT, Institute for Creative Technologies
http://www.ict.usc.edu/ Visited 03 11 20

Kylesten, B. & Hasewinkel, H. (2001) *Lessons learned: Tactical command and control studies with first person shooter games.* Swedish Defence Research Agency, Linköping. FOI-R--0228--SE.

Macedonia, M. (2001) Games, Simulation, and the Military Education Dilemma, The Internet and the University: Forum 2001.

MOVES and Naval Post Graduate School
http://www.movesinstitute.org Visited 03 11 20

SCB, Statistics Sweden, Statististiska Centralbyrån
http://www.scb.se Visited 03 11 20

SIKA, Swedish Institute for Transport and Communications Analysis.
http://www.sika-institute.se Visited 03 11 20

Ubi soft entertainment Information available on the Web: WEBPAGE visited: 2003 09 23

Wikberg, P., Hasewinkel, H., Lindoff, J., Stjernberger, J. & Ericsson, L. (2003) *Game Based Simulation: Using Commercial Game Software in a Ranger Command and Control Exercise.* Swedish Defence Research Agency, Linköping. FOI-R--0989--SE.

The CxBR Diffusion Engine – A Tool for Modeling Human Behavior on the Battle Field

Hans Fernlund, Sven Eklund* and Avelino J. Gonzalez

Intelligent Systems Laboratory
School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816-2450
hfe@du.se, gonzalez@pegasus.cc.ucf.edu

*Computer Engineering Department
Dalarna University
78188 Borlänge, SWEDEN
sek@du.se

Abstract

The option to automatically model the behavior of different actors during live exercise training would increase the value of the after-action-review (AAR) process. If a simulated model of the actors is available right after the live exercise training, the evaluation of their behavior would be more timely and alternative actions could also be evaluated at the same time. The CxBR Diffusion Engine merges technologies to establish a tool for automatic, on-line behavior modeling. Context Based Reasoning (CxBR) is a proven methodology to build simulated agents with human behavior. Genetic Programming (GP) provides the CxBR framework with learning capabilities to automatically create simulated agents with human behavior. The final piece in the CxBR Diffusion Engine is to provide an efficient, flexible, scaleable and mobile platform to evolve the agents' behavior. This platform is the newly developed massively parallel architecture for distributed GP. The massively parallel architecture has the potential to execute the GP linear machine code representation at a rate of up to 50,000 generations per second. Implemented in an FPGA, this architecture is highly portable and applicable to mobile, on-line applications. This paper will present a theory on how the CxBR + GP can evolve simulated agents with human behavior by observation in a massively parallel architecture. These pieces will introduce all the necessary elements to build the CxBR Diffusion Engine that could model human behavior to enable individual AAR of trainees in the training field.

Introduction

The objective of this research is to automatically model human behavior by looking at a human in action. Furthermore, the aim is to either apply these models in a simulator or to use the models to evaluate the behavior of the actor. To model humans in their natural training or operational environment, the automatic modeling tool must be portable. The aim is to have the models ready to use shortly after the monitoring process. To ensure rapid use of the models, the modeling needs to be done on-line. On-line learning is performed at the same time as the actors are performing their tasks. Here we present techniques how this could be accomplished by using a fast executing and portable GP hardware architecture to evolve human performance knowledge in a CxBR framework very quickly and easily. Before describing the technical details, we begin by discussing what could be gained by modeling from observation.

Learning by Observation

Inspired by how humans and other mammals learn by observation, the machine learning community has developed a number of theories on learning by observation, applied in various areas. The interest in this research is to investigate learning human behavior by observation. The intent is not only to use the observations to learn, but also to learn the behavior of the observed entity. A number of advantages could be gained by using learning by observation instead of traditional knowledge acquisition and development methods:

- Reduce time and cost of development, debugging and maintenance.
- More accurate, realistic and refined representation of human behavior.
- Potential to incorporate learning from experience.
- Potential to incorporate new features of human-ness, such as emotions.
- Relaxing operators programming skills.
- Reduction of problem domains.
- Develop simulated entities in real time.

The research described in this paper defines learning by observation as follows:

> *The agent adopts the behavior of the observed entity only through the use of data collected through observation.*

To create a behavioral model with vide variety of human features there must exist an efficient modeling framework. Context-Based Reasoning (CxBR) was proposed by Gonzalez and Ahlers (1998) to have many of these features. Using CxBR as a framework satisfies the prerequisites for implementation of human behavior. If the model is to be created automatically, the framework needs to be equipped with a learning paradigm that will work in conjunction with the framework without disturbing the supported human features. The learning paradigm used in this research is Genetic Programming (GP).

Context-Based Reasoning

Gonzalez and Ahlers (1994) presented Context-Based Reasoning (CxBR) as a modeling technique that can efficiently represent the behavior of humans in intelligent software agents. Later results showed that it is especially well suited to modeling tactical behavior.

CxBR is based on the idea that:

- A recognized situation calls for a set of actions and procedures that properly address the current situation.
- As a mission evolves, a transition to another set of actions and procedures may be required to address the new situation.
- Things that are likely to happen while under the current situation are limited by the current situation itself.

CxBR encapsulates into hierarchically-organized *contexts* the knowledge about appropriate actions and/or procedures as well as compatible new situations.

Mission Contexts define the mission to be undertaken by the agent. While it does not control the agent per se, the Mission Context defines the scope of the mission, its goals, the plan, and the constraints imposed (time, weather, rules of engagement, etc). The *Major Context* is the primary control element for the agent. It contains functions, rules and a list of compatible next Major Contexts. Identification of a new situation can now be simplified because only a limited number of all situations are possible under the currently active context. *Sub-Contexts* are abstractions of functions performed by the Major Context which may be too complex for one function, or that may be employed by other Major Contexts. This encourages re-usability. Sub-Contexts are activated by rules in the active Major Context. They will de-activate themselves upon completion of their actions.

One and only one specific Major Context is always active for each agent, making it the sole controller of the agent. When the situation changes, a transition to another Major Context may be required to properly address the emerging situation. For example, an automobile may enter an interstate highway, requiring a transition to an **InterstateDriving** Major Context. Transitions between contexts are triggered by events in the environment – some planned, others unplanned. Expert performers are able to recognize and identify the transition points quickly and effectively.

CxBR is a very intuitive, efficient and effective representation technique for human behavior. For one, CxBR was specifically designed to model tactical human behavior. As such, it provides the important hierarchical organization of contexts. A full description of CxBR can be found in Gonzalez and Ahlers (1998).

Genetic Programming

Genetic Programming (GP) is derived from Genetic Algorithms and both are stochastic search algorithms. The search process looks for the best suitable program that will solve the problem at hand. The target system for the GP could be a CPU, a compiler, a simulation or anything else that could execute the pre-defined instructions, from now on referred to as a *program*. GP evolves source code representing a program that can address a specific problem. This makes it very suitable for use together with CxBR. GP can build complete software programs that support the internal construction of the CxBR (i.e. the context-base).

To make GP work, some basic requirements must be satisfied. First, we need to have a set of individuals (i.e. programs that represent different solutions to the problem). Furthermore, all the individuals need to be evaluated in some manner as to what degree they are able to solve the problem. The features of the individuals with better suitability would preferably be preserved and survive, or breed new individuals to the next generation. The next GP step would be to evolve the individuals (i.e. reproduction) in some manner to preserve the "good" features and develop even better individuals. The most common genetic operators are *crossover* and *mutation*. They support the development and evolution of the individuals. Evolving a program with GP can be described in three steps, see Figure 1.

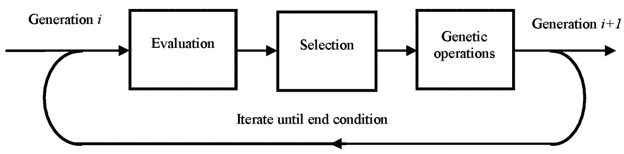


Figure 1: The three steps of GP

The criteria for stopping the evolutionary process can be a maximum number of evaluations made, a maximum number of generations evolved, the fitness reaching a certain level, or other measurable criteria given. When the evolutionary process is finished in GP, there then exists a program that will solve the problem.

Since GP builds source code, it could be used to incorporate knowledge in any context level or in any instances within CxBR where human behavior is encoded. This means that we could choose to implement learning in any specific part of CxBR and construct the knowledge therein.

Towards automation

Human behavior within CxBR can be categorized in two groups: action rules and sentinel rules. Rules, in this case describes knowledge containers. These containers can contain production rules, functions, operators and complex data structures. At the level of any Context, the sentinel rules determine whether their own Context will be active for that specific context level. The structure of these sentinel rules are similar in all contexts at all levels. Each context has its own set of sentinel rules that determine whether this context should still remain active or if it should turn over the control to another context at the same level or at a higher level. This can be viewed as state transition rules where each state (i.e. context) has its own transition table. At the Major Context level, the action set tends to be more a collection of Sub-Contexts and less of other functions, rules or variables. At lower context levels, the action set is less composed of Sub-Contexts and at the lowest context level, there are no Sub-Context calls. Experience has shown that three or at most four, levels of contexts, including the Mission Context, are normally sufficient.

From the discussion above, we can conclude that if we want to incorporate learning into CxBR, the learning paradigm must be able to learn the applicable behavior in a specific context (i.e. action rules) and also the appropriate context switches (i.e. sentinel rules). The type of learning in those two knowledge containers is usually quite different. To learn specific action patterns, the learning regularly becomes somewhat of a regression problem where the model is trying to minimize the discrepancies to the human's performance. When it comes to choose the right context for the current situation, the learning process is more of a classification problem. A learning paradigm that can adapt well to those different learning problems is GP.

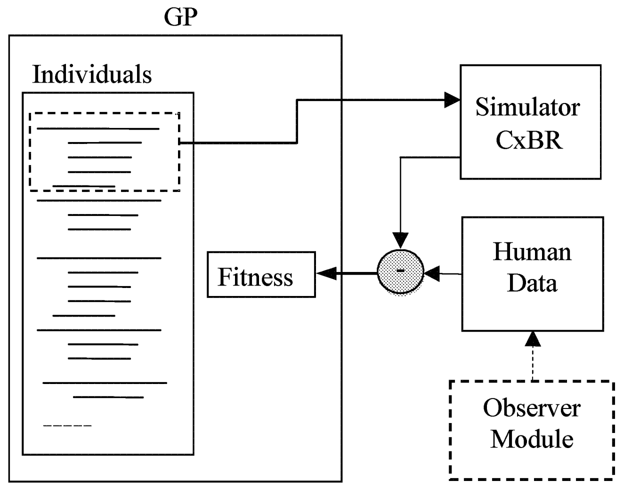


Figure 2: Learning by Observation: CxBR+GP

Instead of creating the contexts by hand, we use the GP process to build the contexts. The GP's evolutionary process provides the CxBR frame with appropriate context's action rules and sentinel rules. The individuals in the genetic population are components of the context base and a simulator is used to simulate an individual's behavior. The behavior from the simulator is then compared with the human performance, and a fitness measure is established to evaluate the models appropriateness. See Figure 2. The evolutionary process will strive to minimize the discrepancies between the performances of the contexts created by GP and the human performance. The human data is the observed human performance, or rather appropriate parts of the performance selected by the observer module. The features of CxBR and GP show that their combination could be a feasible approach to learning by observation. Through this synergistic combination, a system could be constructed that, by observing a human, is able to build a context base for simulated entities that exhibit human behaviors.

The *Observer Module* in Figure 2 has not yet been implemented. In the results described later the objective was to investigate the feasibility of integrating CxBR and GP to achieve our stated objectives. The intended use of the observational module is to select appropriate data to be used in the learning process. Since the computational task was too complex to employ the complete data set this has been done manually so far.

A positive feature of using GP as the learning algorithm is that it preserves many of the features of CxBR that makes it an appropriate paradigm to model human behavior. The tools GP uses to store the knowledge learned is the same tools as a programmer or knowledge engineer would have used developing the knowledge base - source code statements. The agents are able to express their action in a way that is understandable to humans. This implies that the knowledge is easy to interpret if we want to include communicative features within the agents. It is easy to interpret source code and convert it to written language.

Parallel GP

To implement the CxBR+GP concept for constructing human performance models through observation quickly and effectively, we propose a, hardware Diffusion Engine. To achieve this, rapidly on the field, two main issues need to be addressed: mobility and computational complexity. GP is highly computationally complex. To apply GP to a complex problem such as model human behavior and to do it rapidly increase the computational complexity even more. The approach taken here is to present a new massively parallel architecture implemented in hardware to address the computational complexity and yet be able to present a mobile solution.

By distributing independent parts of the genetic algorithm to several processing elements that work in parallel, it is possible to speed up the calculations significantly. This can be done in several ways. The most obvious is to use the independence between individuals, at least during the evaluation step, and evaluate the individuals in parallel. Traditionally, parallel models have also been categorized by the way they handle the population. The choice between a global and a distributed population is mainly a choice on how individuals are accessed and how much communication this will require. The choice between a global and a distributed population is also a decision on GP's selection pressure, since smaller, distributed populations have higher selection pressure, resulting in faster (sometimes premature) convergence.

The Diffusion Model

One of the most interesting parallel models is the diffusion model. This fine-grained model distributes its individuals evenly over a topology of processing elements. It can be interpreted as a global population laid out on a structure of processing elements, where the spatial distribution of the individuals defines the subpopulations or, more appropriate, the neighborhoods.

Every node holds only a few individuals, most often only one or two, and the number of nodes is normally high, making it a massively parallel model with the potential of reaching high speed-ups. The migration in the Diffusion Model is implicit, where fit individuals are allowed to "diffuse" throughout the population.

This is possible since the neighborhoods are defined as the closest proximity around each node over some topology. Since every node is in the center of exactly one neighborhood, these neighborhoods will overlap, making every node part of several neighborhoods. Selection is performed in parallel within these local neighborhoods, and only the center node will be updated in every neighborhood.

In figure 3 the neighborhoods consist of five nodes. During evaluation all nodes evaluate their individuals in parallel. The second step of selection in then done, in parallel, in all neighborhoods. In the example of Figure 3, this means that every node is part of selection processes in five different neighborhoods. However, the selected individual will only update the center node of that neighborhood. Finally, the third step of genetic operations is also done in parallel in all nodes.

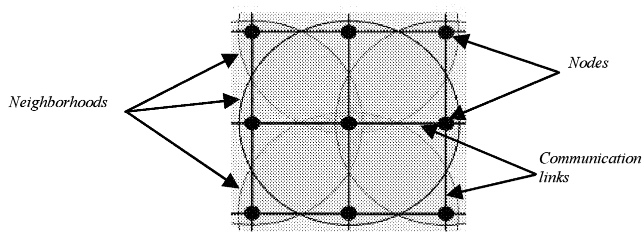


Figure 3: Diffusion Model with a 2D linear topology

An important benefit of the Diffusion Model is that it is well-suited for VLSI implementation since the nodes are simple, regular and mainly use local communication. Since every node has its own local communication links over the selected topology (1D, 2D, etc), the communication bandwidth of the system can be made to scale nicely with a growing number of nodes. Furthermore, the nodes operate synchronously and have small, distributed memories, which also make the Diffusion Model suitable for implementation in VLSI.

In Schwehm, (1996) several other advantages of the Diffusion Model are concluded, most notably the absence of an explicit migration parameter and a potentially higher parallelism than in other parallel models.

The Diffusion Architecture

As noted above, the Diffusion Model is a promising model as such, but it is even more so when considering hardware implementation. VLSI technology, as noted by Kung and Leiserson (1979) during work on systolic arrays, benefits from architectures with small elements, regular structure, local communication, synchronous operation, low I/O and small local memories – all of which are features of the Diffusion Model.

The node

A good starting point when describing the diffusion architecture is to define its basic building block; the node. An architecture based on the Diffusion Model consists of a large number of such nodes. All nodes are identical and are connected to their neighboring nodes over some topology. See Figure 4. They all hold their own unique data (representing their individuals) and they all have the capability to evaluate their individuals and perform the GP algorithm in parallel with the other nodes. See Figure 5.

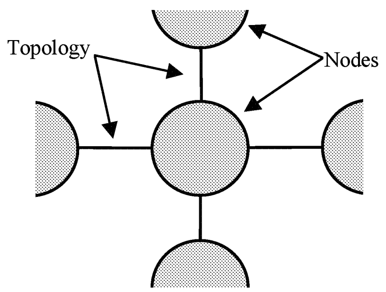


Figure 4: The node in a two-dimensional topology

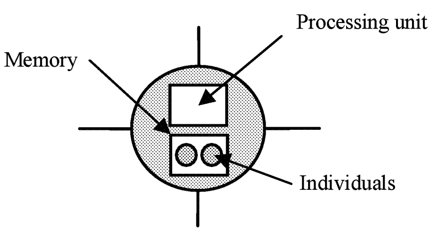


Figure 5: The two major parts of the diffusion node.

Topology

The topology is the “geography” of the communication network in the Diffusion Model - the way nodes are connected to each other. Since fit individuals spread throughout the population over this topology, it is likely that the topology will play an important role in the implementation.

The most common topologies are one- and two-dimensional networks even though higher order ones and tree-like structures are possible. In Figure 6 three common topologies are shown.

Please note that the “edges” of the network of nodes has to be dealt with. Most often the edges are connected “wrap-around” and thus making a one-dimensional line a ring and the two-dimensional grid a donut-shaped toroid.

One of the three major objectives of the design is to make the system scaleable. When implementing the topology in physical chips, the I/O-capability of the chips has to be considered. For instance, implementing 10.000 nodes with an X-net topology in chips that can hold 50*50 = 2.500 nodes would require four chips. Since nodes and their individuals are independent of each other (besides the communication over the X-net), this scales nicely over the four chips that divide the work equally. The wiring needed between the chips, however, is likely to be beyond the capacity of the chips (4 sides * 50 nodes * 3 links = 600 logical links are needed from each chip). Some kind of supporting communication hardware between the chips would probably be needed in this case.

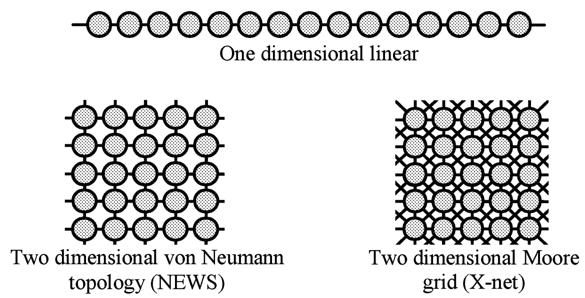


Figure 6: Three “mesh-like” topologies

Neighborhood shapes

Upon the topology of the system, neighborhoods are defined, both by their shape and size. It is within these neighborhoods that selection is performed. Every node is in the center of exactly one neighborhood, but it is also part of, and contributes to, several other neighborhoods, depending on the neighborhood shape. It is only the center node of each neighborhood which is updated by the algorithm.

It is the overlapping of these neighborhoods that enable the implicit movement or diffusion of fit individuals throughout the population/topology. Larger neighborhoods will have a greater overlap and will therefore spread fit individuals faster than smaller ones. Also, using topologies with different dimensionality will spread fit individuals differently.

The distributed GP

In the distributed population model of the Diffusion Model the basic steps of the genetic algorithm from Figure 1 are performed in parallel in all nodes. Since the population is distributed and communication between individuals is limited, these basic steps are somewhat different than in the standard GP.

Fitness evaluation

The node and the representation of its individuals have to incorporate the ability to evaluate very different individuals even though all the nodes have to obey identical control signals from the single control unit. This could be solved by either a very general representation (data parallel evaluation), by an embedded evaluation unit under local node control or by a combination of both.

Selection algorithms

Traditional selection algorithms such as roulette or ranking selection require global fitness or ranking averages to be calculated, distributed and maintained. Implemented in a parallel model, this often introduces a communication bottleneck which will limit the performance of the parallel system. By using a local selection pool only, that is, the local neighborhood of the Diffusion Model, this communication problem can be managed satisfactorily. However, this also means that the standard GP is changed for most selection algorithms and it is not obvious how this will affect the overall performance of the system.

Genetic operations

Mutation is a stochastic change of a single individual, independent of all other individuals. It can therefore be performed in parallel over all nodes without any communication.

Crossover is the recombination of two individuals and may therefore require some communication, depending on the population density and on the selection algorithm. The implementation is dependent on representation but also on the type of crossover.

Representation

The three steps described above are all heavily dependent on the choice of representation. The choice of representation for the individuals is also central in order to achieve the main objectives of effectiveness, flexibility, scalability and mobility.

Since the evaluation of the individuals is responsible for a large portion of the total execution time (for most applications), the representation has to be fast to interpret and evaluate. It has also to be storage efficiently ("hardware friendly") since fewer transistors per individual means more individuals (or nodes) per chip which most often translates to shorter overall execution time. Fewer transistors per individual could also be translated into smaller size with the same number of individuals (nodes), which benefits the objective of mobility. Finally, the representation has to be flexible enough to efficiently support many different applications with minimal changes to the design.

We propose the use of linear machine code as GP-representation. Clearly, it is storage efficient and it is very efficient to evaluate for a CPU. Using general purpose registers it also supports reuse of results and easy parameter passing.

Since the code and therefore also the behavior differ between individuals, each node has to have its own resource to evaluate its individuals (which basically is the whole point of a parallel system). Using linear machine code GP it is natural to equip every node with its own CPU.

This also means that the system can quite easily adapt to new applications. The solution space of the application (the space of all possible solutions) is simply defined by the Instruction Set Architecture, ISA. By redefining the CPU and its ISA the architecture can be fine-tuned for each application. Using a hardware description language like VHDL, this is easily accomplished.

A general structure of the node architecture is shown in Figure 7.

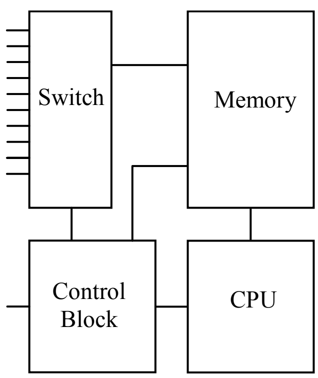


Figure 7: The four blocks of the diffusion node

Experimental Results

The validity of combining CxBR and GP to learn human behavior only from observational data will be shown in Fernlund's upcoming dissertation. The new approach to learning by observation was applied to simulated automobiles. The experiments collected data from humans operating a simulated automobile and the new approach evolved a stable agent, with all behavior pattern learned, able to operate in a simulated environment. One of the most interesting results was that the automated model was as good as a model created by traditional means (i.e. manually by an independent knowledge engineer/programmer).

The Diffusion Model has been extensively studied through simulations in order to define some of its most important parameters, for instance selection algorithm, topology and neighborhood size. Please refer to Eklund (2003a) for a detailed description of these results.

The Diffusion Model has also been tested on several benchmark applications. In Eklund (2003b) the model proved to work well on the very difficult task of handwritten character recognition. In Eklund (2003c), the model was shown to outperform some of the most well-known models for time-series forecasting.

Recently, a first prototype of the diffusion architecture has been implemented in VLSI. Preliminary results indicate that one node of the diffusion architecture, holding one individual, can be implemented using roughly 30,000 gates. Using a symbolic regression problem as application and a maximum individual size of 32 words, the entire population can be updated at a rate of 50,000 generations per second, regardless of population size.

Conclusions and Further Work

In this paper we have presented the necessary theories to create a rapid, on-the-field human behavior modeling tool. Results have shown that GP and CxBR are able to evolve simulated agents with human behavior, solely by observations. The new Diffusion architecture has shown very good simulation results and the first hardware implementation is currently being built. The theories presented here fulfill the prerequisites for enabling the creation of tool that could be used to rapidly build human behavior models on the field.

Even if the different parts show good results, further research is needed to make them work together. First the CxBR and GP module needs to modify the tree structure representation of the individuals to suit the linear machine code representation of the diffusion architecture. To make the theory of learning by observation complete, which also will make the approach more generic and widely applicable, further research needs to be conducted within the observation module. This module will completely automate the modeling by observation and analysis of the environment. It is also necessary to develop the observer module to enable on-line learning. Furthermore, additional adjustment is probably needed from both hardware and the algorithmic side to complete the CxBR Diffusion Engine.

Completing the CxBR Diffusion Engine would open possibilities of model human behavior and automatically create simulated agents with the same behavior, without the observed human being aware of it. In certain situation it could be preferable to not declare the modeling to the actors prior to the experiment. The modeling could also be done in a live event and not during a simulation. The possibility could be to even model hostile troops. We could also use the models in After Action Reviews and either use the models in simulated actions or evaluate the knowledge stored in the models. Since the knowledge created by the GP module is transparent, it is possible to understand and use the knowledge learned.

References

Eklund, S, "Empirical Studies of Neighbourhood Shapes in the Massively Parallel Diffusion Model", Advances in Artificial Intelligence, Lecture Notes in Artificial Intelligence no. 2507, ISBN 3-540-00124-7, Springer-Verlag, 2003a.

Eklund, S, "Handwritten character recognition using a massively parallel GP Engine in VLSI", IFAC Conference on Intelligent Control and Signal Processing, Faro, Portugal, April 2003b.

Eklund, S, "Time series forecasting using massively parallel genetic programming", The Fifth International Workshop on Nature Inspired Distributed Computing, NIDISC'03, Nice, France, April 2003c.

Gonzalez A. J. and Ahlers, R. H. "A Novel Paradigm for Representing Tactical Knowledge in Intelligent Simulated Opponents" Proceeding of the IAE/AIE Conference, May 31-June 3, 1994, Austin, Texas.

Gonzalez A. J. and Ahlers, R. H. "Context-Based Representation of Intelligent Behavior in Training Simulations" Transactions of the Society for Computer Simulation International, volume 15, no 4, December 1998

Kung, H. T, Leiserson, E, "Systolic Arrays (for VLSI)" in I. S. Duff and G. Stewart (editors), Sparse Matrix Proceedings, Knoxville, Academic Press, 1979.

Schwehm, M, "Parallel Population Models for Genetic Algorithms", Universitat Erlangen-Nürnberg, 1996.

M&S and Artificial Intelligence

Improving Computational Efficiency in Context-Based Reasoning Simulations

Roberto C. Sanchez, Avelino J. Gonzalez
University of Central Florida
Orlando, FL
sanchezr@ieee.org, gonzalez@ucf.edu

Abstract

The current application of the competing context approach to solve ambiguities in context-based reasoning (CxBR) simulations has limitations. This paper describes several improvements to the current competing context approach that allow a more dynamic and and preemptive evaluation of the simulation environment. In their previous work, Saeki and Gonzalez developed the idea of the competing context as a method to choose a course of action from among two or more possibilities in CxBR simulations. Essentially, the concept is that the simulation halts while the intelligent agent generates a list of possible context transitions and, if necessary, enters a "time-warp simulation," selects each context in a "what if" approach to see what will happen, and then makes a selection based on the "best" outcome. The major drawbacks to this approach are the lack of realism introduced by relying on random simulated events as part of the decision making process and the computational burden as the number of choices grows. To effectively cope with a large number of possible choices while still exhibiting realistic tactical decision making, the CxBR framework can be improved. The generation of situation interpretation metrics (SIMs) is made more robust by incorporating degraded states, resulting in a much richer representation of the simulation environment. The more robust representation of the simulation environment allows the agents to be more proactive in decision-making, thus improving the computational efficiency of the simulation and increasing the realism by more closely reflecting actual human tactical behavior.

Introduction

Over the past several years context-based reasoning (CxBR) has developed from simple scripts layered over a rule-based system (Gonzalez & Ahlers, 1993) to a full-fledged framework (Norlander, 1999) capable of supporting a complete simulation environment (Gonzalez, Grejs, & Gonzalez, 2000), resolving ambiguous situations (Saeki & Gonzalez, 2000a), and allowing very fine-grained control of the simulation environment (Gallagher, Gonzalez, & DeMara, 2000). Gonzalez and Ahlers have also shown that the CxBR approach is particularly well-suited to the representation of human tactical behavior (1998), allowing for the development of simulations that maintain a high level of fidelity because the intelligent agents in the simulation behave as a human is expected to in the same situation.

While the CxBR framework is quite powerful in its current form, it quickly consumes all avialable resources on a reasonably powerful personal computer. Gonzalez, Grejs, and Gonzalez observed that even with modern computer hardware, a medium sized driving simulation with more than five or six agents begins to slow quite dramatically (2000). The slowdown stems in part from the manner in which the agent in the CxBR simulation decides which course of action to take when more than one valid choice is available. To solve this predicament, Saeki and Gonzalez implemented the competing context concept as a way to allow an agent to decide which is the "best" course of action by invoking a "time-warp simulation." However, as the number of valid possible courses of action increases, either from more available choices or from the presence of additional agents, the simulation experiences a significant slowdown (Saeki & Gonzalez, 2000b).

Brézillon, Pasquier, and Pomerol (1999) showed that in cases where different decisions were simply variants of each other, the separate branches could be combined to reduce the number of choices, as long as the order of the steps themselves was unimportant and the result was the same. While this approach provides gains in process oriented environments, such as the subway system application that Brézillon, Pasquier, and Pomerol developed, tactical environments are naturally more diverse. For example, a tank that suddenly encounters the enemy must choose between attack, defend, evade, and retreat. The contextual graphs developed by Brézillon, Pasquier, and Pomerol do not provide the needed performance gain.

The best way to boost the performance of the CxBR simulation without compromising the integrity or fidelity of the simulation environment is to improve the competing context approach by making the process more dynamic and evenly spread across the simulation. Taking a dynamic approach also has the effect of eliminating the "time-warp simulation" in many cases where it would normally be used, and of improving the situation interpretation metrics (SIMs) to allow more effective categorization of the situation.. This research proposes modifications to the CxBR framework that allows the simulation to better track and evaluate the data associated with each agent and with the simulation environment as a whole, so that at any moment each agent can know exactly how any particular decision (based on the locally available information) would impact its own situation.

CxBR Overview

Assumptions of CxBR

According to Gonzalez and Ahlers (1998), CxBR is based on the following assumptions:

- "Tactical experts are proficient in their tasks by recognizing and treating only the key features of the situation."
- "There are only a limited number of things that can realistically take place in any situation."
- "The emergence of a new situation will generally require alteration of the present course of action."
- "Tactical knowledge is quite general in nature."

Essentially, CxBR assumes that the expert is, in fact, an expert and that he knows what he is doing with respect to his area of expertise. Thus, an auto mechanic would be expected to troubleshoot a car engine but know little or nothing of performing brain surgery. While someone's level of proficiency is a rather subjective matter, it is reasonable to expect that the individual would not be considered an expert if he or she lacked the fundamental skills to effectively deal with situations which encompass the area of knowledge. The expert is also expected to separate necessary and unnecessary information as part of the decision-making process. For example, when troubleshooting a car engine, the fact that the upholstery is cracked and fading is largely irrelevant—an expert is expected to identify such information as irrelevant.

While it is possible to surmise that an unlimited number of things can occur in any period of time, it is not realistic. This is not to say each situation is limited to a very small number of happenings, but rather, that the situation dictates what should be expected. An auto mechanic troubleshooting an engine would expect only a small number of things to happen, all of which would be at his discretion, such as connecting the engine to a diagnostic computer, starting the engine, and checking the various parts. Likewise, a military commander leading a tank platoon in an engagement against an enemy tank platoon would expect occurrences from a limited range of possibilities. Thus, the situation itself tells the expert what to expect, and the expert is expected to glean this information upon entering the situation.

An expert typically takes a particular course of action based on the current situation and the available information. Had the expert entered into a different situation, it is entirely possible that the expert would have elected a different course of action. So, when the situation itself changes into a different situation or when a different situation comes into play, it is expected that the expert will change his course of action to reflect his evaluation of the new situation.

General Description of CxBR

The CxBR paradigm represents human tactical behavior by defining a hierarchy of contexts that can direct the AIP's actions. Essentially, CxBR is effective and efficient because it uses the currently selected context to exclude irrelevant knowledge from the decision-making process (Gonzalez and Ahlers, 1998). A good illustration of excluding irrelevant knowledge is that of an automobile driver on the highway—the driver does not need to consider knowledge related to traffic lights, only those things which can conceivably occur on the freeway need be considered.

The way that CxBR is composed, each agent is controlled through its own set of contexts (Gonzalez & Ahlers, 1998). At the highest level, the mission context defines the goals, constraints, and other parameters relevant to the agent. The agent is controlled through a major context and the various sub-contexts. The mission context contains several major contexts that the agent can assume to perform various tasks. For example, a tank may have the following main contexts available: patrol, attack, defend, retreat. The major contexts are exclusive of each other, so that a tank cannot attack and retreat at the same time. Beneath each major context are several sub-contexts, which act as helpers to the major context. The sub-contexts may or may not be exclusive, such as when an attacking tank is maneuvering to avoid enemy fire and reloading the main gun at the same time.

By controlling when and how each agent transitions into and out of each context, it is possible for the agent to behave just as an expert would in the same situation. This system would be ideal if all context transitions were known at development time and if they were small in number. However, as Saeki and Gonzalez (2000a) observed, in tactical situations all the valid context transitions cannot normally be identified in advance and, if they could all be identified, there are far too many to realistically hard code them. To overcome this obstacle Saeki and Gonzalez developed (2000a) and implemented (2000b) the competing context concept. [For a more complete discussion of CxBR please see (Gonzalez & Ahlers, 1998), (Norlander, 1999), (Gonzalez, Grejs, & Gonzalez, 2000), (Saeki & Gonzalez, 2000a), and (Gallagher, Gonzalez, & DeMara, 2000)]

Overview of the Competing Context Approach

Saeki and Gonzalez assert that the difficulty in predefining all possible context transitions in a tactical situation proves difficult because of the number of variables that must be considered for each decision (2000a). They developed the competing context concept as a way to evaluate several potentially valid context transitions and select the "best" one. The determination of which context is "best" lies with the fulfillment of the goals or needs of the agent. For example, a tank may have the goals of (1) survival, (2) protect friendly units, and (3) defend a hill. Any action the tank takes (in the form of a context transition) should serve to maximize the goals it has at that time.

The competing context approach itself has four distinct steps (Saeki & Gonzalez, 2000a):

- Situation interpretation metrics generation
- Relevant context group selection
- Context attribute matching
- Time-warp simulation

Situation interpretation metrics (SIMs) provide a way for the agent to evaluate the current situation against its current goals. The SIMs can be anything relevant to the simulation environment and are only limited in number by the available memory. Some examples of SIMs in a military simulation are: current state of the agent, number and location of friendly units, number and location of enemy units, weather, and terrain features.

Relevant context group selection occurs when the agent's current goals are matched with the context's attributes. Each context that possibly meets the goals of the agent is added to a list for later consideration in the "time-warp simulation." This is the first step in reducing the number of available choices for the context transition.

Context attribute matching is a preliminary elimination of the relevant contexts that have been selected. Any context attribute which is not present or does not meet the predefined criteria can be safely eliminated. For example, to enter an attack context the friendly unit must be within range of an enemy unit. If the enemy unit is out of range, or there is no enemy unit present, the attack context can be eliminated from consideration. This is the second step in reducing the number of available choices.

In the event that the previous three steps result in more than one available choice, the main simulation is halted and a "time-warp simulation" is started. The "time-warp simulation" follows the transition to a valid context, executes the prospective context, and evaluates the resulting SIMs. This procedure is repeated for each possible context transition. The SIMs for each execution are compared to determine which context transition best fulfills the AIP's goals, with that context being declared "best." The simulation then returns to normal execution and the selected "best" context is activated.

Limitations of the Competing Context Approach

While the implementation of the competing context approach did much to resolve the ambiguities common to tactical limitations, the approach is limited. In the event that the time-warp simulation results in a situation where a nested time-warp simulation becomes necessary. Rather than execute a nested a nested time-warp simulation, one of the available contexts is simply chosen at random (Saeki & Gonzalez, 2000a). Another limitation is that the competing context approach has only been applied to the context transitions at the major context level, and does not currently extend to the sub-context transitions because of the computational expense.

The random nature of the simulation also makes it possible for the same situation to result in significantly different context transitions (Saeki & Gonzalez, 2000b). Because a random function is used to determine certain occurrences, such as a hit or kill, the results of the "time-warp simulation" may differ from one simulation run to the next. For example, if the agent is faced with a decision about whether to attack the enemy or take some other action, the first run of the "time-warp simulation" may result in a killing blow. The agent then decides that because the action is successful, in the "time-warp simulation," to attack. However, the next time the agent is faced with the same decision, the "time-warp simulation" may result in a missed shot causing the agent to select another course of action. Of course it must also be considered that the way the "time-warp simulation" occurs does not indicate how the same action in the same simulation will occur. This is perhaps the greatest limitation of the competing context approach.

Improvements to the Competing Context Approach

The limitations of the current competing context approach notwithstanding, it is a very powerful way to increase the flexibility and applicability of CxBR throughout the realm of computer simulations. We propose several improvements to the current competing context approach that would make it more computationally efficient, more representative of human tactical behavior, and foster extending it to more completely encompass the entire CxBR framework. The desired extension can be gained by making the following improvements:

- Incorporate the degraded states model (Gallagher, Gonzalez, & DeMara, 2000) into SIM generation
- Make relevant context group selection a dynamic and continuous process by anticipating what will occur
- Make context attribute matching a dynamic and continuous process to match with dynamic context group selection
- Eliminate the time-warp simulation in most instances in favor of a preemptive evaluation of the situation

The incorporation of the degraded states model into SIM generation allows a higher degree of control over the agent's actions. Making context group selection and attribute matching dynamic activities can be accomplished by using spare CPU cycles during periods of "down-time" in the simulation. The modifications to the first three steps then make it possible to begin evaluating potential new situations before they occur.

SIM generation with degraded states

The work done by Gallagher, Gonzalez, and DeMara on degraded states modeling showed that the realism of an agent's behavior could be increased by providing more fine-grained control of the agent's individual state (2000). This was accomplished by allowing individual aspects of the agent's state to take on anyone of a range of discrete values. For example, rather than a tank being simply alive or dead, its capabilities were divided into separate categories, such as fuel level, ammunition level, damage sustained, and crew fatigue. Dividing the agent's status into individual components allows the agent to make more competent decisions, and thus appear to more closely reflect human behavior.

The degraded states model can easily be extended to the SIMs themselves. Instead of having a SIM that is *number and location of enemy units*, it is possible to divide that information into separate pieces and incorporate some of the enemy unit's degraded state information. The additional information contained in the SIM would be available to all agents in the simulation, so it would be necessary to ensure that only information which could be available to all agents is included. For example, if a tank's turret were destroyed, this degraded state information can be made available to all agents in the simulation since a simple glance at the tank would reveal this. However, other information, such as fuel level, would not be made available since there is no way to know the fuel level of another vehicle without direct access to it. Thus, an agent that is completely out of fuel would appear as *immobile*, without further elaboration to other agents in the simulation.

The SIM generation should also be accomplished continuously so that at any point in time the agent has all the necessary information available with which to make a decision. This is not only more efficient, by spreading the SIM generation to less computationally intensive times in the simulation, but also more accurately reflects human tactical behavior.

While the generation of SIMs with degraded states does not itself constitute an improvement to CxBR or competing context, it does provide a much more robust representation of the simulation environment. The added robustness is necessary to facilitate the preemptive evaluation of the situation.

Dynamic relevant context group selection

In the current competing context model, when the agent reaches a point where a context shift must occur and there are multiple choices available, the selection process begins by creating a list of possible contexts to which a transition is possible. This action should take place as part of any changes in the agent's goals, rather than when it is time to make a decision.

There are several times throughout the simulation when the agent's goals can change, including (1) immediately following a context shift, (2) because of a change in the simulation environment (including an update to the SIMs), or (3) from an outside stimulus. In each case, the opportunity exists to reevaluate the relevant context groups. If each available opportunity were taken to update the list of relevant contexts, then when a context shift is necessary, the selection is already complete and the overall computational expense of this activity has been more evenly distributed over the simulation.

Performing relevant context group selection dynamically not only reduces a computational bottleneck found in the original competing context approach, but it more accurately represents how an human expert would continuously evaluate a changing situation. This also paves the way to further improving the competing context approach by allowing preemptive evaluations of possible context transitions.

Dynamic context attribute matching

Once the list of relevant contexts has been updated, the context attributes can be matched, eliminating context shifts that are not possible. Like the relevant context group selection, this can be accomplished throughout the simulation without the need to wait for a point where a decision is necessary. Also like the relevant context group selection, the overall computational expense is evenly distributed over the entire simulation.

A natural time in the simulation to match context attributes is whenever the relevant context group has been updated. Like the dynamic relevant context group selection, this has the effect of more accurately representing human tactical behavior. This process essentially allows the agent to select from a small list of likely context transitions that can then be evaluated against the current state of the simulation.

Preemptive evaluation of the situation

By making the context group selection and attribute matching dynamic and continuous, the dependency on the time-warp simulation is dramatically reduced. In the event that a context shift becomes necessary and a context has not been unambiguously preselected, the time-warp simulation can still be used a fall-back.

The rationale behind the preemptive evaluation approach is that it more accurately represents human tactical behavior. A human expert constantly evaluates a changing situation and anticipates likely occurrences. Only when an unanticipated situation occurs is the expert forced to stop and consider additional options.

For example, an agent (a tank) may be patrolling an area that is a known minefield and where the enemy has not been sighted for two weeks. The agent has the following contexts to which a transition is possible: *attack-enemy*, *retreat*, *evasive-maneuvers*, *maneuver-and-return-fire*, and *cover-wingman*. Since the area has no known enemy forces and is a minefield, the agent can conclude that it is likely that he or his wingman could strike a mine. Given the current situation, the agent can evaluate the situation where his wingman strikes a mine (before it even happens) and decide that because there are no enemy forces present, he will remain and enter the *cover-wingman* context. As long as the assumption that no enemy forces are present holds true, this preemptive evaluation is valid and need not be changed.

As the simulation progresses, if the wingman actually strikes a mine, the situation would normally trigger a context competition. However, because the situation has already been evaluated, the only thing that happens is that the simulation verifies that the relevant assumptions made at that time still hold and then proceed to execute the context transition.

If an enemy tank appears, an occurrence which is not anticipated, then as a last resort the agent is still able to perform a normal context competition. However, this only occurs because the encounter with the enemy was not expected, as the available information indicated that it was not likely. This approach more closely represents human tactical behavior because is mimics the expert's continuous evaluation of a changing situation and the need to consider additional options to cope with an unexpected situation.

Summary

The proposed improvements to the competing context approach represent a significant improvement in the computational efficiency and realism of CxBR simulations by utilizing the improved competing context approach. The improved approach to competing context also solves the problem of making a random choice in the event of a nested context competition (Saeki & Gonzalez, 2000a), because if the context transition has been anticipated there is reduced need to enter the time-warp simulation, and the possibility that a nested context competition will occur diminishes very quickly. In the worst case, all anticipated context transitions are invalid and the simulation must fall back to the original context competition by evaluating the situation when the need for a decision presents itself. Thus, nothing is lost and much can be gained.

Current Status of Research

The improvements to the competing context approach have been defined. The CxBR framework also already exists as developed by Norlander (1999), and the competing context approach has been integrated by Saeki and Gonzalez (2000b). Thus, the current code base must be modified for the implementation of the proposed improvements. The improved model may change slightly in response to any unforeseen difficulties or considerations during implementation. Once the improvements to the competing context approach are implemented, experimental data need to be acquired. The data can be used to compare the performance and realism of simulations utilizing the original competing context approach as compared to the performance and realism of simulations utilizing the improved approach.

Future Work

Because, in addition to improving the computational efficiency, the improved competing context approach improves the realism and representation of human tactical behavior, extensions to this work may find applicability in the area of neural networks.

References

Brézillon, P., Pasquier, L., & Pomerol, J.-C. (1999). Modeling Decision Making with Context-Based Reasoning and Contextual Graphs. Application in Incident Management on a Subway Line. *Proceedings of the Human Centered Processes Conference*, 331-336.

Gallagher, A., Gonzalez, A.J., & DeMara, R.F. (2000). Modeling Platform Behaviors Under Degraded States Using Context-Based Reasoning. *Proceedings of the Interservice / Industry Training Systems and Education Conference, 2000*.

Gonzalez, A.J., & Ahlers, R.H. (1993). Concise Representation of Autonomous Intelligent Platform in a Simulation Through the Use of Scripts. *Proceedings of the Florida Artificial Intelligence Research Society Conference*.

Gonzalez, A.J., & Ahlers, R.H. (1998). Context-Based Representation of Intelligent Behavior in Training Simulations. *Proceedings of the Naval Air Warfare Center Training Systems Division Conference*.

Gonzalez, F.G., Grejs, P., & Gonzalez, A.J. (2000). Autonomous Automobile Behaviour Through Context-Based Reasoning. *Proceedings of the Florida Artificial Intelligence Research Society Conference*.

Norlander, L. (1999). A Framework for Efficient Implementation of Context-based Reasoning in Intelligent Simulations, (Master's thesis, University of Central Florida, 1999).

Saeki, S., & Gonzalez A.J. (2000a). Soft-Coding the Transitions Between Contexts in CGF's: The Competing Context Concept. *Proceedings of the Computer Generated Forces and Behavior Representation Conference*.

Saeki, S., & Gonzalez A.J. (2000b). Competing Context Concept: Experimental Results. *Proceedings of the Interservice / Industry Training Systems and Education Conference*.

Implementation of a Prototype Context-Based Reasoning Model onto a Physical Platform

Viet C. Trinh, Brian S. Stensrud, Avelino J. Gonzalez

Intelligent Systems Laboratory
University of Central Florida
Orlando, FL
vtrinh@isl.ucf.edu, brian@isl.ucf.edu, gonzalez@ucf.edu

Abstract

This is SAWMAS Context-Based Reasoning (CxBR) is a intuitive paradigm for human behavioural representation, presenting a new and unique method for modelling autonomous intelligent agents. This modelling paradigm is based on the simple fact that knowledge is modularized into contexts, which contain all of the intelligence needed to successfully address particular situations. CxBR models have shown promise in recent exercises within simulated environments. To date, these simulations have involved tactical missions undertaken by submarines, tanks, and other military vehicles as well as driving missions undertaken by various automobiles. This paper discusses the first attempt at integrating this modelling paradigm in a physical platform - specifically the implementation process and the many issues addressed when translating this model onto a live agent. In a simulation, CxBR models execute in a controlled environment, with controlled inputs (i.e. sonar, positioning) to the agent. The major problems encountered with porting this agent onto a physical platform are caused by environment unpredictability, input noise, and agent capability degradation. An iRobot ATRV-Mini agent entity was used in this integration. Due to the basic functionality of the ATRV-Mini, a very primitive CxBR model was developed. The model, in no sense, presents the full potential of the modelling paradigm nor the iRobot itself, but instead illustrates the ability of integrating a CxBR model onto a physical platform. This paper will include the details of the scenario used to prove the feasibility of controlling physical entities through CxBR, it will also describe the evaluation made of the exercises. Although this task seems fairly simple in nature, it becomes a challenging task to integrate the CxBR model to the physical robot. This integration marks the first attempt to incorporate a CxBR model into a real-world scenario, and its success paves the road for integration of much larger and sophisticated models on live physical platforms. Furthermore, this proof-of-concept demonstration is a springboard for applying and evaluating new CxBR research - such as collaborative CxBR models, the competing context concept, and the automation of the CxBR model development process using machine learning or knowledge acquisition tools - on not only simulated but also physical autonomous platforms.

Introduction

This is In the past decade, Context-Based Reasoning (CxBR) has been applied in various research projects. It has established itself as a successful paradigm for human behavioural representation of intelligent autonomous agents, as shown in its many publications. The initial concepts of CxBR started as scripts implemented by Gonzalez and Ahlers [Gonzalez & Ahlers 1993], and enhanced by Norlander's CxBR Framework [Norlander 1999], the first attempt at constructing an engine for CxBR. Current research in CxBR includes learning, competing contexts, planning, and collaborative behaviors (teamwork). All previous research has been in tactical applications in the military-related domain, with the exception of automobile driving. It involves the control of one or more autonomous agents in a simulated environment. Simulations can be used for training of military personnel and testing of scenarios with simulated vehicles. Although simulations are useful in training and analysis, models can also be put to practical use in the real-world. There is more control of the environment in simulation than in real world situations. Real world situations are unpredictable, input data can be noisy and the actions taken by the agent are not perfect. Although noise can be added to simulations modelled after the real world, it still remains a well controlled environment. Therefore the idea of implementing and testing a CxBR model on a physical platform is necessary to fully evaluate the potential of CxBR. Military vehicles as well as automobiles are designed to function in the real world, giving greater importance in integrating artificial intelligence models into these physical platforms. This paper provides an introduction to CxBR, describes the physical platform (an iRobot), the scenario used for this endeavour, the implementation process and methods, and the results obtained from the study.

CxBR Overview

Context-based Reasoning (CxBR) is "a novel behaviour representation paradigm that can effectively and efficiently be used to model the behaviour of intelligent entities in a simulation." [Gonzalez & Ahlers 1998]. This paradigm is designed to be used for human behaviour representation, utilizing the use of contexts as a reasoning method, similar to how humans make tactical decisions [Gonzalez & Ahlers 1998]. CxBR models have shown success in recent exercises within simulated environments. To date, these simulations have involved tactical missions undertaken by submarines, tanks, and other military vehicles, as well as automobiles on the road. CxBR is based on the idea that [Gonzalez & Ahlers 1998]:

- "Any recognized situation calls for a set of actions and procedures that properly address the current situation."
- "As a mission evolves, a transition to another set of actions and procedures may be required to address a new situation."
- "What is likely to happen under the current situation is limited by the current situation itself."

CxBR is focused on the concept that humans think in terms of contexts. [Norlander 1999]. Contexts encapsulate knowledge/intelligence about appropriate actions needed to address specific situations [Gonzalez & Ahlers 1998]. The context controls the behaviour of the autonomous agent, assigning it appropriate actions based on the active major context in which the agent is situated. The CxBR paradigm is composed of a tactical agent, a mission context, major contexts, sub-contexts, and sentinel rules.

Tactical Agent

This is the agent that is completely controlled by the CxBR model. It is given a mission to execute, and all of the allowable contexts required to accomplish the particular mission assigned to it. Actions the agent performs are controlled by the CxBR model, specifically the contexts within the model.

Mission Context

The mission context has no direct control of the agent, It is a high-level specification mechanism for the entire scenario. The mission context defines all of the parameters used for the scenario and the agent itself (i.e., terrain data, scenario boundaries). The objective of the mission is contained in the mission context, along with the list of applicable major contexts [Gonzalez & Ahlers].

Major Context

A major context encapsulates all of the knowledge required to execute a certain task, based on the current situation. The context contains all of the functions, rules, and compatible context transitions for that context, as well as a list of all applicable sub-contexts [Fernlund & Gonzalez 2002]. Only one major context is in control of the agent at any one given time-this is called the active context. There is always exactly one major context active. Defined in the mission context are the default and initial major contexts. The default context is the major context activated when no applicable contexts available. The initial context is the active context with which the mission begins.

Sub-Context

A sub-context is a representation of low level procedures that are auxiliary to major contexts [Gonzalez & Ahlers]. A major context can contain many sub-contexts to assist the context in its execution. Sub-contexts can also belong to many major contexts, this allows reusability in CxBR [Gonzalez & Ahlers].

Sentinel Rules

Sentinel rules are those monitoring rules that dictate that a transition to another context is necessary. Sentinel rules constantly monitor the condition of the agent and its environment. Whenever a context is no longer valid and a transition is necessary, a sentinel rule is fired and a context switch is initiated.

The CxBR Framework developed by Norlander [Norlander 1999] allows the integration and development of various CxBR models. The framework performs all of the lower level actions required of CxBR, (i.e., maintaining states, transitions, initializing contexts, etc.) allowing the developer to concentrate on generating the mission, the contexts, the agent interface, and the agent itself. The implementation in this paper utilizes this framework to execute the CxBR models.

iRobot Physical Platform

The physical platform used as a host for integrating CxBR is iRobot's ATRV-Mini. The ATRV-Mini is an all terrain robot designed specially for research purposes, ideal for this project. It is the smallest of the ATRV series, therefore the simplest to implement. This paper will makes references to the iRobot ATRV-Mini as simply "iRobot". The iRobot is controlled by the Mobility™ software, residing in its onboard computer [iRobot Corporation 2003]. The iRobot's computer runs on a Redhat Linux 6.2 operating system. Some of the key features of the iRobot are [iRobot Corporation 2003]:

- Battery Operated: 2 - 4 hours
- Centralized, general purpose onboard computer
- Breezenet wireless network
- 16 Sonars (6 front, 8 side, 2 rear)
- An internal odometer
- Joystick control

The wireless network and the battery operation allows the iRobot to freely roam around a terrain (within the wireless network's range) and establish a constant connection to a client computer. The iRobot's centralized onboard computer allows remote access through SSH as well as programming and compiling directly on the iRobot. For this case, the CxBR program controlling the iRobot can reside onboard, eliminating the problem of integrating a networked application. The robot is capable of moving forward and backwards, with 360^o of rotation. While moving or rotating, the iRobot's internal odometer maintains its relative position and heading. The sonar enables the iRobot to detect objects in its immediate surroundings. The mobility software allows the iRobot to detect objects up to four meters away. However, through field testing, the iRobot is only capable of detecting objects approximately two meters away. The iRobot has built-in C++ header files that allow a CORBA connection to the Mobility™ software that controls the iRobot. A full list of features can be obtained from the official iRobot Web site (http://www.irobot.com) [iRobot Corporation 2003]. Details on the integration are presented in the Integration section.

CxBR Mission Scenarios

The scenario put forth for this paper is a simple tactical mission. The mission is designed as a proof of concept, and has no practical utility. The mission is a basic scouting mission is which the iRobot moves around a terrain in search of an enemy entity. Sub-contexts were not used because of the simplicity of the mission and contexts. The proof of concept is to integrate a CxBR model onto an iRobot and analyze its performance.

Mission Objectives: The objective is to maneuver an iRobot around an open area looking for a single enemy entity. Upon detection, determine the hostility level of the enemy. If it approaches, consider it hostile and retreat. If the enemy retreats, follow it at a close distance. If the enemy is not responsive (i.e., stationary), execute an end of mission signal and retreat to the original starting position.

Mission Constraints:

- Flat 8m x 8m surface (paved parking lot).
- One enemy present, no other objects.
- A predefined wait time
- Objective is to detect a stationary enemy.

Context Set:

- locateEnemyContext - Maneuver the iRobotaround the field searching for the enemy. Moving from waypoint to waypoint. (*initial and default context)
- determineEnemyHostilityContext - After the enemy is detected, the iRobot determines the enemy's hostility by waiting for it to move in either a retreating or advancing direction. If the enemy is not moving, it is considered stationary. After an approach or retreat maneuver, the enemy's action is re-scanned therefore transitioning back to this context.
- retreatContext - If the enemy is approaching, retreat moving backwards.
- approachContext - If the enemy is retreating, approach the enemy slowly. (Half the distance between the iRobot and the enemy).
- stationaryEnemySignalContext - Signal (using a
- signature maneuver) that the enemy has neither retreated or advanced for certain period of time, then conclude mission by leaving the area and going back to its original position.

Context Methodology

The five contexts or the context-base that is implemented in simulation assumed ideal iRobot actions and a perfect environment to interact with. In the case of simulation, these aspects can be made ideal and perfect, however, in the real-world these factors can not be controlled as well or at all. As simple as the mission and implementation may be, the real world model needed many modifications to adjust the contexts and AIP to these inconsistencies and inaccuracies experienced with the iRobot and its environment. Below is a description of how each of the contexts is implemented. Figure 1 illustrates all of the legal context transitions.

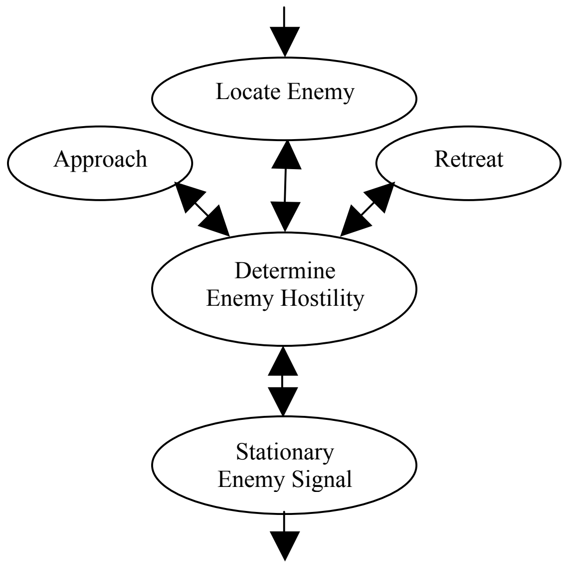


Figure 1: Context Transition Diagram for Scouting Mission.

locateEnemyContext

Originally, the iRobot was supposed to move and rotate simultaneously to achieve a more circular and continuous movement scheme. Because of the terrain, it was difficult to perform both actions at the same time while remaining accurate. Therefore rotating and moving were separated to reduce possible inaccuracies. The iRobot now moves from waypoint to waypoint, moving in a straight direction and turning only at waypoints. Because the iRobot moves and rotates to a tolerance, the position and heading might be somewhat inaccurate. Performing one straight movement to the waypoint would cause the iRobot to possible miss it because it would never be with in the certain range tolerance. Therefore its movement is broken down into small straight movement sections with minor rotations to compensate for the iRobot being off course.

determineEnemyHostilityContext

This context is used to determine the enemy's actions and how the iRobot should react to the current situation. This context is transitioned to whenever the enemy's action needs re-evaluation. In order to accurately determine the enemy's hostility level, the iRobot remains stationary. Whenever the enemy moves, the iRobot's uses its internal odometer and sonar to analyze the enemy's movement and position. If the enemy is moving closer, it is considered approaching and the retreatContext is call upon. If retreating, then the approachContext is called upon. In order to detect a stationary enemy, the enemy has to remain within a certain tolerance area when detected. This tolerance area allows to compensation of sonar fluxations.

retreatContext

When an enemy is approaching, it natural for a human to turn around and retreat. However, in vehicles, it is more advantageous to advance backwards because turning is quite slow. If turning is performed to quickly, the vehicle might flip over. The iRobot has sonar in the rear allowing it to have limited visibility of the enemy while moving backwards. Originally, the iRobot would only move a small distance away from the enemy and then perform another detection cycle on it. This process was too slow and the iRobot was susceptible to collusion with the oncoming enemy. Therefore the iRobot quickly moves three meters backwards completely avoiding the enemy.

approachContext

Whenever the enemy is retreating, the iRobot will approach slowly with reduced speed. Ideally, the iRobot would trail the enemy robot until it becomes offensive and advances on the iRobot. However, dealing with speeds, distances, heading, and timing simultaneously on the iRobot was very difficult. Therefore, the iRobot would approach the enemy by moving half way between its position and the enemy's position. Sometimes the enemy can retreat very fast and drop out of the iRobot's sonar. This is will shift the context back into locateEnemyContext.

stationaryEnemySignalContext

This context performs a signature move for the iRobot by rotating in one direction, then rotating in the other direction. This was the simplest context to implement because it did not involve the enemy. The iRobot was to move back into its original starting position to measure the level of inaccuracy caused by executing the mission.

Figure 2 shows the diagram of the scouting mission scenario. This scenario shows the iRobot in the locateEnemyContext.

Implementation Process

The implementation process was broken down into three steps: Developing the CxBR model on the current CxBR framework, development of a CxBR to iRobot interface, integration of the CxBR model with the iRobot. After implementation testing, refining, and validation of the model occur.

CxBR Model Development

The initial CxBR model was developed on a separate machine running an identical operating system to that of the iRobot (Redhat Linux 6.2). This eliminated discrepancies between the development platforms and

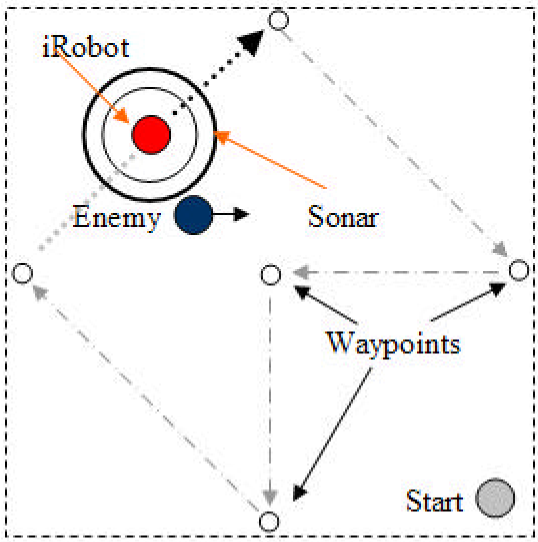


Figure 2: The Scouting Mission Scenario layout with the iRobot in the *locateEnemy* context.

compiler versions. The CxBR framework is build to run on various platforms [Norlander 1999]. Within the CxBR model, the iRobot entity is represented by an agent class called AIP (Autonomous Intelligent Platform). This agent class performs all of the high level functions of the iRobot utilized by the contexts. The contexts classes were then implemented, calling upon the iRobot agent to perform actions. Within the contexts classes are the sentinel rules for the contexts and the valid context transitions. The mission class is then implemented with the agent and all of it respective contexts. The models developed were run using a dummy robot agent interface to validate of the functionality of the model.

iRobot Interface

The iRobot is implemented as an inherited class (iRobot AIP) from the base AIP class. Within the workings of the iRobot agent class (iRobot AIP), the iRobot interface is called upon by this class to allow the framework, more particularly, the contexts used by the framework, to control of the iRobot through its functions. The functions to control the robot require a CORBA connection to the mobility software running on the iRobot. The iRobot interface performs the low-level functions of the iRobot (i.e. communicating with the mobility software), while the iRobot tactical agent performs the high-level functions. The commands to move and rotate the iRobot were the next important when programming the interface. The command to move the robot only accepts the speed in meter/sec (positive-forward direction, negative-reverse direction) and not a position to move to. However, in the scenarios implemented, a distance/position is required. To fix this problem, the odometer was used. The robot was given a certain fixed speed to travel and the iRobot positioning on the odometer was constantly monitored until the desired position was reached within a certain tolerance. The same problem occurred with rotation, instead of translational velocity, angular velocity and the odometer heading was used. The iRobot was given a certain fixed angular velocity to rotate, and the heading on the odometer was constantly monitored until the desired heading was reached, again within a certain tolerance. The sonar is used to detect objects in close proximity of the iRobot. There were sixteen sonar and the data collected by the each sonar was the x (front) and y (side) distance from an object. If the absolute distance is four meters (the range of the sonar) then it is assumed there is nothing in front of that particular sonar. However, the relative detectable distance used is two meters. The stop command was issuing a zero speed command. These are the low level command implemented by the interface along with monitoring and reporting the iRobot's internal odometer and sonar readings.

CxBR Model Integration

The integration of the model with the iRobot required porting the CxBR model, framework, and interface directly onto the iRobot itself. After the establishment of the iRobot interfacing, the iRobot AIP, along with the CxBR software, the model was ready for integration. Figure 3 is a diagram of the components used for the integration, and the flow of control from the CxBR model to the physical iRobot. The iRobot is ultimately controlled by the contexts implemented in the CxBR model. However, the contexts use high-level robot control functions implemented in the iRobot agent class. The iRobot agent in turn calls lower level functions implemented in the interfacing code. The interfacing code calls the mobility software built into the iRobot's internal computer with movement and sonar level commands. The mobility software is what controls the iRobot's motor and sonar functions. The integration process began by porting the CxBR model from simulated iRobot AIP to the physical iRobot AIP implementing the iRobot interface.

Testing

The perfect world/agent assumption did not prove true in the real world as it could in simulation. It was quickly noticed during testing that the iRobot had inaccurate movement, rotations, and sonar readings. To the iRobot itself and the mobility software, the movement and sonar readings are perfectly executed. The iRobot did not know of its own inaccuracies. However, as the developer, the inaccuracies were apparent by observing the iRobot in action. Therefore the iRobot interface and the iRobot agent code were refined to address these particular issues. The refinement of the model was fine tuned to allow this margin of error in movement and sonar detection to occur while maintaining adequate performance levels. The agent

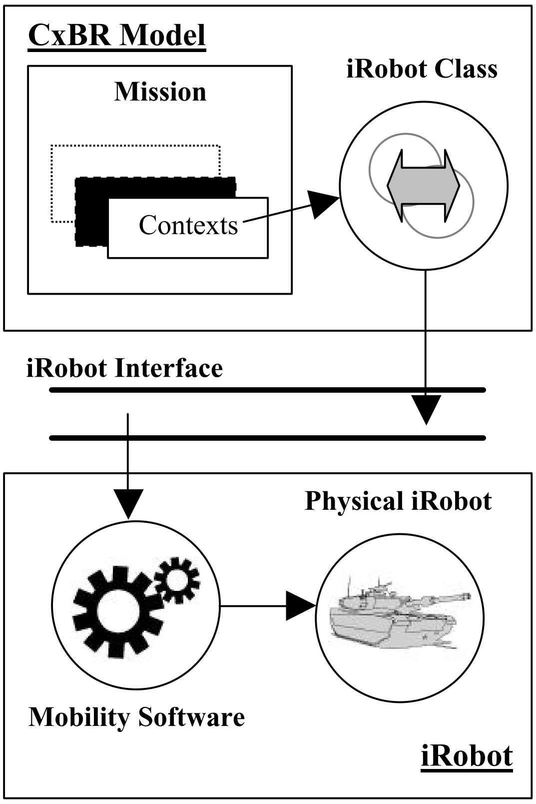


Figure 3: The Integration Process of the CxBR Model onto the iRobot Platform.

actions were modified to partially compensate for these inaccuracies. Compensations were introduced in the low-level iRobot interface. The purpose of testing was to determine whether the CxBR model was proficient enough to adequately compensate for the inaccuracies of the iRobot and its function as well as its performance in the real world. Five runs for scouting scenario were used to test the capabilities of the CxBR model on the robot.

The test metrics used for testing were:

1. A sequence of events is plan for the enemy. Does the iRobot react appropriately to the emeny actions?
2. Does the model correctly transition contexts? Are these transitions distinctly visible during the execution of the mission?
3. Does the model complete its mission successfully?
4. How did the iRobot telemetry and sonar inaccuracies affect the performance?

These test metrics are implemented to analyze the proficiency of the CxBR model to execute the specified missions. Testing showed how CxBR was able to cope with a real world environment.

Results

After extensive execution and refinement of the CxBR model, the results proved very interesting. The iRobot is able to execute its missions and the contexts are visible during the execution. However, the context transitions are not perfect. Some transitions are completely missed at times when a transition is required, as well as incorrect transitions made due to the degraded capabilities of the iRobot and the physical environment. Below is an explanation of the results. There were timed pauses made between transitions in the CxBR model. This provided the framework a small time period where it was able to analyze the situation and transition properly. If the time gap is removed, the sonar would read very similar values before and after enemy evaluation, possibly causing incorrect transitions to occur. This time delay mechanism was integrated in the refinement stage to allow time to differential between movement and error. These pauses also constitute a time-step in which the iRobot performance can be evaluated. Performance evaluation based on the test metrics:

1. The iRobot is able to detect the enemy whenever it comes within the sonar radius of the iRobot. However, as the battery life degrades the enemy has to come closer to the iRobot for detection to occur. When the enemy moves very close to the robot representing a hostile action, the iRobot is not able to detect such close movement. When the enemy is approaching (hostile) or retreating (not-hostile) the iRobot is able to detect the majority of these occurrences and reacts to the enemy by retreating and approaching respectively.
2. For the locateEnemyContext, the transitions are executed properly. During normal operations of locateEnemyContext, the iRobot is constantly moving from one waypoint to another. When an enemy is detected, the iRobot immediately halts and transitions to determineEnemyHostility-Context. The determineEnemyHostilityContext is the most problematic of all. This context connects all other contexts together and therefore has the most work to perform. Sometimes the iRobot would not detect the robot moving closer or further from it and remains in determineEnemyHostility context, ignoring the need for a context switch. Sometimes the iRobot thinks the enemy is lost and begins locate-EnemyContext in search for the enemy when the enemy is still nearby. To prevent problems with context transitions in the retreat and approach contexts, an automatic context transition is made at the end of each execution of the actions (retreating or approaching) back to determine-EnemyHostilityContext. However, there were no incorrect transitions to invalid contexts, except for

transitioning to locateEnemyContext when the enemy is still close. Is occurs when the enemy drop off the sonar readings. This is a problem of the physical iRobot, and not of the CxBR model. To the CxBR model, all transitions are valid.

3. The iRobot as able to detect a stationary enemy and complete its mission by executing the stationaryEnemySignalContext.
4. The iRobot's sonar inaccuracies greatly affect the performance. Sometimes, there are misreading in the sonar and the enemy disappears off of the sonar map, thus causing an incorrect transition to locateEnemyContext. The telemetry inaccuracies have a minor affect on the performance. The CxBR model is still able to correctly transition and execute contexts with telemetry error. Only the actions are only minimally affected.

Overall, the iRobot is able to perform the mission. It encounters difficulties in the determineEnemyHostility-Context and the inaccuracies in the telemetry and sonar reading. Determining the difference from a not-in-range, stationary, hostile, and non-hostile enemy action can be confusing to the iRobot at times. Because of this time gap introduced between contexts, the iRobot may not be able to react to all of the enemy's actions. However, with the compensations added, these errors are rare, and the iRobot is able to perform proficiently in its environment.

Summary

The iRobot is a well-developed platform for testing the validity of the implementation of a CxBR model onto a physical platform. Overall, the model was successful in controlling the iRobot through the missions. However, there were some problems faced. The main problem with the iRobot was the inaccuracy of its internal odometer and the sonar. Within the scenario, one of the end-of-mission actions is for the iRobot to return to its original (0,0) position and face forward at 0^o. When the iRobot returned to its original position, the iRobot was positioned incorrectly by a half to one full meter in the x and y directions. The direction the iRobot faces is between 5^o to 10^o off of the original position. This inaccuracy was also detected by observing the iRobot constantly move out of its programmed boundaries. This error is not related to the software, but instead to the physical machinery of iRobot itself. The problem mainly arises from the tires of the iRobot and the terrain the iRobot is in. The testing was performed in a large parking lot with an asphalt pavement. When the wheels move on the pavement, the iRobot loses some traction and the actual positioning is different from the desired and internally measured position. Internally, the iRobot thinks it travels a certain distance, but in reality it has moved less than the desired distance, creating an inaccuracy in the odometer. Therefore, every movement the iRobot performs can further skew the odometer reading. The sonar inaccuracy occurs slowly with time. Since the iRobot is battery powered, the sonar's detection radius decays with the time the iRobot is operating in the field. This affects how the iRobot detects the enemy. With degraded sonar capabilities, the enemy has to move closer to the iRobot to be detected, and a collision may occur, limiting the time for testing.

Future Work

The scenario used in this integration was simple. It does not portray the full robustness of the context-based reasoning paradigm. Integrating a more complex CxBR model would be the next step in this research. Because of the limitations of the current iRobot platform, better robotic platform will be used to create this model. Learning from observation is another topic which uses this robotic implementation. With the ability of porting CxBR from simulation to real world, learning can be performed by observing experts maneuvering an agent in the real world, building a CxBR model through the knowledge acquired, test it in simulation, and port the model back onto the original agent. One major extension to this work should include the development and integration of a full CxBR model onto a complex physical platform.

References

Fernlund, H.K., & Gonzalez, A.J. (2002). Automatic creation of intelligent simulated entities through context based reasoning and genetic programming. *Proceedings of the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC).*

Gonzalez, A.J., & Ahlers, R.H. (1993). Concise Representation of Autonomous Intelligent Platform in a Simulation Through the Use of Scripts. *Proceedings of the Florida Artificial Intelligence Research Society Conference.*

Gonzalez, A.J., & Ahlers, R.H. (1998). Context-Based Representation of Intelligent Behavior in Training Simulations. *Proceedings of the Naval Air Warfare Center Training Systems Division Conference*.

iRobot Corporation (2003). http://www.iRobot.com

Norlander, L. (1999). A Framework for Efficient Implementation of Context-based Reasoning in Intelligent Simulations*, (Master's thesis, University of Central Florida)*.

Adaptive Planning Concepts using Emerging Flexible Manufacturing Simulation Methods

Fernando G. Gonzalez
Intelligent Systems Laboratory
University of Central Florida
fgonzale@ucf.edu

Abstract

Flexible manufacturing is a relatively new concept in manufacturing where the plant is given many different types of products in very small lot sizes to manufacture. This type of highly flexible manufacturing plant is dynamically changing at a must faster pace than traditional manufacturing plants. As such new highly adaptive planning algorithms have been studied in the attempt to utilize the full flexible capabilities of these systems. Many military applications require the same type of adaptive planning that is used in flexible manufacturing. This paper explores the parallelism between adaptive planning in manufacturing and in military applications. It specifically looks at a new technique that gives rise to multi-resolutional decision-making. This technique allows for decisions to be made at all levels of the control hierarchy, with different amounts of resolution using the same set of models. It represents a big savings in the effort of creating the models and maintaining them up to date with the current information. The paper shows how this concept can be used in several military applications. As a decision support system it can analyze or even suggest different courses of actions (COA). Given the models of the different military units and the different options available to the commander this system can suggest the best COA. If the models are maintained up to date with the latest situational information it can respond quickly to plan a new COA.

Introduction

This paper addresses a Decision Support System (DSS) that can provide networked collaborative decision support to aid in the planning, coarse of action (COA) analysis, resource management, and mission execution. Our conceptualized system can provide distributed, multi-resolution, fast and accurate decision making capabilities while minimizing the workload in creating and maintaining the system's models. This system can support the war fighters by either evaluating a particular COA or suggesting a COA. The DSS system can use simulation so that a decision can be made automatically without human intervention. This allows for the DSS to respond with a suggested COA quickly. The DSS is maintained with up to date information about the current military situation and is continuously searching for a better COA. It is capable of dynamic re-planning across geographically separated echelons. Since the decision the system produces must be intelligent the system models the opposing force as being intelligent. This is essential to an automatic decision-making process since not accounting for the enemy's intelligence may lead to a vulnerable COA.

In addition to evaluating a particular COA the system can be capable of suggesting a COA. The system can have the capability to accept the specifications of options for various decisions. The system will then search through the solution space made from the combination of options to find a COA that performs best according to a specified criterion. A COA consists of a selection of an option for each decision. The DSS system uses the fact the military uses a hierarchical command structure to make its decisions. This hierarchical command structure combined with our distributed modelling methods allows for easy selection of modelling resolution without further modelling. It also provides decision support at all levels of command hierarchy.

Related Work

The following research efforts are related to our work. Peters et al. (1996) [19], Smith et al. (1994) [21] and Smith and Peters (1998) [22] have adapted Arena Kelton, Sadowski and Sadowski (2001) [17] to control their experimental FMS. However, unlike the modelling approach to be discussed in this paper, their control architecture is limited to only one hierarchical level where a single supervisor, the cell controller, manages a set of subordinate processes. In their modification of Arena, they have included special events in order to facilitate message passing among the controllers. Furthermore, to model a complex FMS or other multi-level, distributed system, the set of modelling elements provided by ARENA, as well as most simulation languages, severely constrains the modelling process. This is particularly true when one attempts to assess the impact of the control architecture upon the system. Davis et al. (1993) [9] and Davis (1998) [6] details the numerous restrictions that current simulation languages impose upon the modeling of hierarchically distributed systems. Mize et al. (1992) [15] further discusses the inaccuracies that ensue from using current simulation languages in order to model an FMS.

Narayanan et al. (1992) [16], Govindaraj et al. (1993) [11] and Bodner and Reveliotis (1997) [4] have also developed simulation tools that are capable of controlling a system. They claim that the typical approach to address the complexity of FMS control systems is a hierarchical decomposition. A widely used decomposition features tree layers, Strategic Decisions, Tactical Decisions and Operational Decisions, Arango and Prieto-Diaz (1991) [3] and Stecke (1985) [23]. In [4], [11], and [16], they claim that to handle lower level issues involving the real-time control, two additional layers are needed. The Structural and Equipment Control layers address issues such as deadlock avoidance. In these papers, hierarchical decomposition refers to the logical decomposition of the decision making process while we refer to it as the actual decomposition of the controller that yields a distributed controller. Incidentally they also claim that FMS exist but there are no controllers for them.

The notions of multi-resolutional architectures and task decomposition have been discussed by many. The interested reader is referred to Albus and Meystel (1995 and 1997) [1] and [2] for a discussion of these terms. In the same reference, these authors discuss their reference architecture for distributed intelligent control and the associated real-time control system for its implementation. The major distinction between their architecture and ours, discussed below, is that they have separated the planning and control functions within a given controller. Our approach does not separate these functions. Secondly, we make heaver use of on-line simulation.

Optimization is an area where much effort has been placed. Optimization in the context of manufacturing plant performance includes the scheduling of resources. The most widely studied scheduling problem is that of assigning a task to each machine at each time. Reyes, Yu and Lloyd (2001) [20] presents a method where they use a Genetic Algorithm (GA) search along with a Petri-Net (PN) model to eliminate the infeasible solutions by using it in the fitness function. They search for an optimal selection of a process plan for each job in the system. Kacem, Hammadi (2002) [14] uses a method called Approach by Localization and they combine it with GA to schedule each task on each machine. They not only determine the task each machine will have but also the time that each task will occupy each machine. Dimopoulos and Zalzala (2000) [10] present a survey of the various scheduling approaches. While these scheduling approaches do an excellent job of scheduling the tasks on the machines, they only look at the scheduling problem isolated from the manufacturing system. That is, they only consider the machines and the tasks they can perform without regard to other constraints set forth by the manufacturing plant. For example nearly all of the formulations of the scheduling problem, irrespective of the approach that is being adopted to solve the problem, ignore constraints associated with material handling and the providing of supporting production resources such as tooling and fixtures. They also almost always neglect the machine setup times as well as the inherent uncertainties that exist in the system.

Because this technology is not yet mature to be implemented in a real time controller a different method is being considered that is much easier and gives good schedules although it may not yield the optimal solution. It is based on optimizing the dispatching rules for each queue in the system. A dispatching rule is an ordering of the tasks that are waiting to use a particular resource. Examples are first in first out and earliest due date. Panwalker and Iskander [18] present a list of 100 dispatching rules commonly used.

While much work has been done in the planning or scheduling of manufacturing plants, much less work has been done in the physical control of manufacturing plants. The issue of physically distributing the intelligent controller among many computers is not being addressed in the literature. There have been efforts to use the same model for both control and simulation, see Gonzalez (1996) [12] and Smith and Peters (1998) [22]. This gives the simulation high fidelity and makes the state transfer from the control model to the simulation model trivial. However when the fact that the controller needs to be distributed is considered, the other existing methodologies falls apart. This is due to the fact that the controller will use a collection of models while the simulation uses only a single model. With the current research, the models cannot be the same. Consider the additional complication resulting from the need to transfer the state from a collection of models to a single, different, model

The Distributed Modelling Methodology

Naturally, for the DSS system to work, the operation must be modelled. Since a military operation consists of many individual ships, airplanes, soldiers, other personnel and so on and is distributed over a wide area, a single model running on a single computer will not have the capability to handle the demand of the DSS system. For this reason the model must consists of a set of many smaller models. The system must accept a collection of independent models for consideration.

For example consider a hypothetical organization with the hierarchical command structure shown in Figure 1. The organization has a commander that is at the top. Under it are 5 units that report directly to the commander. Each unit has itself a unit commander and 3 sub-units that report to its unit commander.

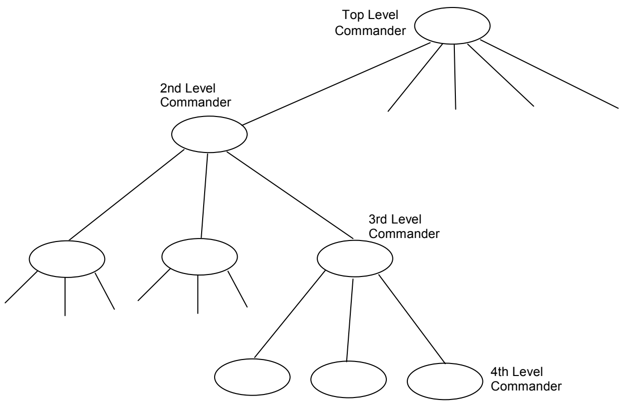


Figure 1. A Sample control hierarchy for our hypothetical system.

The topmost commander makes high level long term decision while the middle level commanders make shorter term decisions. The units in the lowest levels make local immediate decisions. If one were to simply build a model for each decision maker, the top level commander will have a model that incorporates its own logic as well as the logic for the lower units and, depending on the resolution desired, may also include the logic for the 3rd and 4th level units. The 2nd level units will have a different model. This model will include the logic for the unit itself plus the logic for the units under it. The lowest level unit's model will only incorporate its own logic. The current approach is to build a model for each decision maker to use. The topmost commander will use a model that includes the logic of each of the 5 units under it as well as the logic from the lower level units. The 5 units in the 2nd level will each have a model including its own logic. Note this logic appears in 2 different models. With many units, each making decisions, there will be a model for each commander. Each model will contain the logic that is involved with the decision. That is, all of the units except those of the lowest level will contain its own logic as well as the logic of the units below it.

Under our approach each unit will have a unique model that only includes its own logic. While their will still be a model for each commander however since each model only includes its own logic there is no model duplication. Now, when the top most commanders need to make a decision, it uses its own model that incorporates its own logic and, to include the logic of its sub units, it includes the models that correspond to those lower level units as well. Now the model is not a single model but rather a collection of models. In order to use a collection of models, the simulation engine must be able to accommodate a set of models rather than just one as is the case with the current state of the art in simulation.

Taking knowledge from the control of flexible manufacturing community, we have designed and tested a simulation approach that allows a collection of independent models to be used to describe the system. Since our system is designed to control a manufacturing plant it is can actually control the system by making decision, then issuing commands to implement the decision and finally monitor the systems performance. Although this control aspect of our design will not be appropriate in a military application it serves as an indication of the approaches potential in making decisions.

This approach gives rise to multi-resolution decision-making. The commander can vary the resolution of the simulation by varying the number of models included in the simulation. The more models that are included the higher the resolution but the more processing required in the simulation. If the commander wants a fast low resolution decision it may only include the first 2 levels in the hierarchy. A higher resolution model will include models at lower levels as well.

One of the greatest benefits to this approach is the model reusability. When the top most commanders execute a simulation run it may include all of the models in the levels below it. At the same time the 2nd level commander uses its own model as well as the models in the lower levels. Notice that these models are being used by the top most commander as well as be the commanders at the lower levels. In fact any commander that wants to execute a simulation simply selects the models that it wishes to use and includes only those models. By having one model per unit with no duplication of the logic in other models, all decisions are made with a consistent understanding of each unit's logic. This gives rise to higher fidelity predictions.

Model reusability is an important issue since the models not only have to be created but also maintained current. This not only includes modifying the models when the structure of the organization changes but more importantly maintaining the models up to date with the latest state information. The state of a model is basically a series of variables that holds information pertaining to the dynamically changing situation. Examples include the current weather, the location of resources such as troops, tanks etc., the political situation and so on.

A decision is only as good as the amount of information that was used in making the decision. A good decision must consider the current state of the system. Any decision support system that is capable of evaluating and suggesting different COA must maintain the current state of the system in order to be ready to make good decisions. This allows for the dynamic re-planning. The goal is to minimize the effort in maintaining the models current. By having one model per unit with no duplication of the logic in other models, the portion of the state information that corresponds to that model is used to update only that model. Without using our approach possibly many models will need to be updated with the state information corresponding to single unit. This is because the same logic is included in several models.

Given a model initialized to the current state, a simulation using this model is capable of predicting the future outcome of the system under the current state. The DSS system is capable of suggesting a COA. Given all of the decision options available to the user our system will vary those options in order to search for the best COA. Since there are typically a very large set of possible options Genetic Algorithms (GA) may be employed to search through this large space of options for the best solution.

Another requirement for a good decision is that it must consider the COAs that the enemy may take. The enemy must be modeled and be part of the simulation. The enemy must be assumed to be intelligent and therefore a good decision must continuously searches for the best COA for the enemy as well. Therefore simulation must incorporate any action the enemy may take. The system will also use the same intelligence that it uses to find the best COA for the user to find the best COA for the enemy.

The Simulation

Discrete-event simulation is used to give an understanding of how a system will perform in the future under the conditions set forth in the model. A discrete-event simulation is concerned with the discrete event aspects of the system, those aspects whose state changes by discrete increments at discrete times. A military decision is considered to be a discrete event problem since a decision consists of choosing a particular COA among a large but finite set. In fact any decision that is not concerned with the continual monitoring and regulation of a continuous process such as a motor is considered to be discrete event in nature.

Existing research considers the simulation and control of a discrete event process. Control is performed by issuing commands to the physical system while simulation is used to make intelligent control decision. The DSS system is not intended to control a military operation but rather simply to find and suggest the best COA under the current situation. However a simulation consists of a controller / system pair were the physical system is not involved. Since the aspects of the system that the simulation is controlling in the models are discrete, the physical system is modelled using delays. For example in a real system we issue a command to the real system to perform some task then wait until the system completes this task. At this time the task-complete event is executed. In simulation the same basic process occurs but the task-complete event is scheduled to occur at a selected time in the future. This selected duration models the delay that the real system takes performing the task. The duration time is a random variable taken from a distribution that appropriately models the actual delay. The following section describes the method of using a collection of models for simulation versus a single large model.

The following section describes the existing methodologies that will be use. It is assumed that the controller consists of a collection of controllers distributed across many computers and tied together by the network. Davis (1992), [5] Davis et al. (1993), [9] Davis et al. (1996 and 1997) [7], and [8] and Gonzalez and Davis (1999) [13] present the architecture for the controller and the single threaded simulation of distributed systems using distributed modelling. They present a solution to the problem of controlling a flexible manufacturing system by developing a framework for the intelligent control of flexible manufacturing systems and other distributed DESs. This methodology is based upon the belief that the interactions among the controllers must be considered by the simulation model in order to accurately model a system with a distributed control architecture. The single most important characteristic of the methodology and what separates it from other object-oriented simulation approaches is its attention to modelling the flow of messages among the controllers included within the architecture. By modelling the flow of messages, the methodology allows the simulation to accommodate distributed models.

The System Model

The executive object is the object that runs the simulation. It assigns a unique event number for each possible event that can occur. When it calls the system model implemented as a C++ function, it passes the event number. The model function uses this event number to determine which sub-function of the model to execute. Thus, as defined here, the model function simply represents a large case statement or switch statement with the event number being the switch variable. Because the time it takes the controller to process an event is negligible when compared to the time that the system requires to perform a typical task, time does not advance while executing an event.

Logically, an event occurs when an entity residing in the system needs to change its state. For example, a units has completed the assigned step and needs to be told to begin the next step in the COA. Our simulator changes the state of the entity by executing events as specified in the model, until the entity can no longer change its state without either waiting for some resource or waiting for a given task to be completed.

The simulation is built using modeling elements. These modeling element are the building blocks that are combined to build larger more complex models. Currently our tool has two basic modeling elements. These modeling elements are used to model the waiting periods that occur in the real system. The QUESEIZE modeling element is used to model the waiting period that occurs as a result of an entity waiting in a queue for one or more required resources. In a military sense this can be used to model the waiting of certain events to occur before proceeding with the plan. The DELAY element models the delay period involved in the execution of task. This model the time it takes for the unit to complete task it was commanded to do. Other modeling elements can be created and added dynamically in the process of building the model. The architecture of the simulation tool allows for easy modeling element creation and addition. Every application whether it is manufacturing or decision support requires a unique set of modeling element.

Discrete-event systems usually have entities flowing among their physical elements whose flow must be both explained and coordinated. This may be the flow of troops on the ground, the flow of airplane going from a base to another and so on. In the simulation these entities are represented with a C++ structure. The entity can accommodate a serial number, locations to store attributes that describe its state, a list of elements to store the COA or plan if it is appropriate for that entity, and an address stack that stores the return address for program-control purposes. This return address defines which segment of code the program should return to once the currently selected segment of code has been processed and is used to provide subroutine calls in the model.

The Simulation Engine

The following is a brief description of how our simulation methodology is able to accommodate a set of models in its simulation rather a single model. The simulation of each individual subsystem is performed using the commonly used discrete-event simulation method. An executive function manages the simulated time, event list and the list of resources and calls the model (the single model) to execute events as they are pulled off of the chronologically ordered event list. The simulated time is incremented to the time of the next event in a single discrete jump. All of these components, the event list, the list of resources, current time and the model, are encapsulated into a single object, called the simulation object. See Figure 2. In this way, each simulation object simulates a single subsystem using the standard discrete-event simulation.

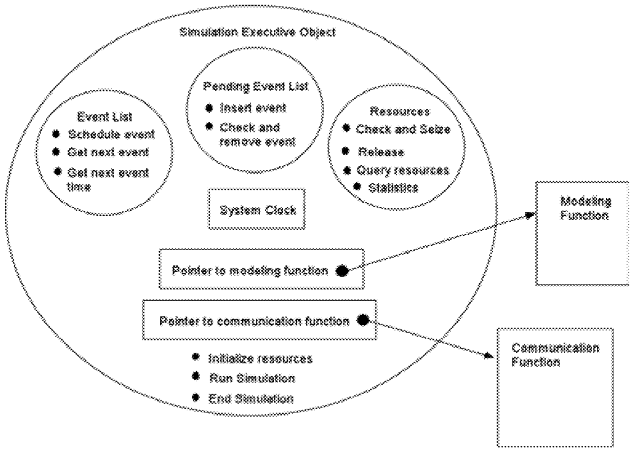


Figure 2. The Executive Object. The circles represent programmed objects, the squares represent functions, and the flat rectangles represent variables.

In order to simulate the entire distributed system, the overall simulation model must include an instantiation of this object for each modelled subsystem. A coordinating function for the overall simulation manages the execution of each simulation object. This function resembles an operating system in the sense that it distributes the processor usage among all of the simulation objects. This function allocates the processor to the simulation object that needs to execute the next event in time. It maintains a global event list that has a copy of each event in each of the simulation objects. It also models the network by maintaining a list of all messages sent between the simulation objects. This message queue is also used to determine which simulation object acquires the processor next.

As illustrated in the example in Figure 3, the coordinating executive function removes the next event, the one occurring in controller number 3, from the global event list. It gives the processor to simulation object number 3. Object 3 executes this event and as a result of its modelling logic it sends a message to object number 2. This message is sent to the message relay that simulates the network. Once object 3 completes its cycle it relinquishes control back to the coordinating function. The coordinating function then removes the next message from the message queue and delivers it to the proper recipient, object 2 in this case. Next object 2 receives this message and executes the appropriate event. In the execution of this event, controller number 2 schedules a new event onto its local event list and a copy is scheduled onto the global event list. At this point, the event-execution cycle of the coordinating function terminates and the function removes the next event from the global event list and begins a new event-execution cycle.

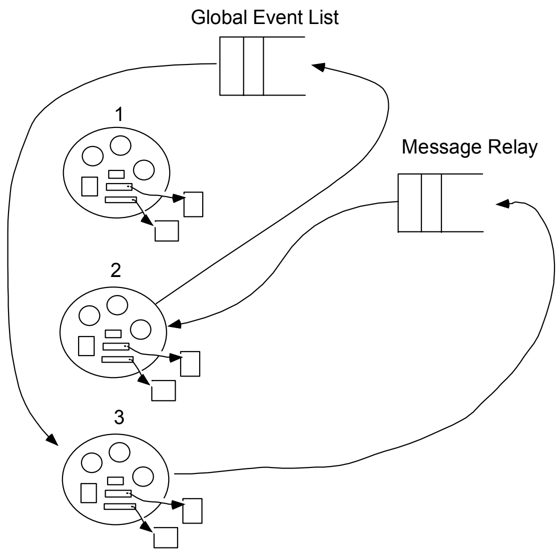


Figure 3. Example of a HOOPLS simulation of a control architecture comprising three controllers. Each of the three controllers is simply included in the overall model as a simulation object without any modification.

The System State

The current system state, or current situation, is used to maintain each model up to date with the current military situation. The state contains information pertaining to any aspect of the system that is continuously changing. Examples include the weather, political situation. Location of troops and other resources, current military policies, amount of resources such as food, water, fuel and so on. Things not included in the state but hard coded into the model are those that do not change over time such as the maximum velocity capability of a particular tank or the control architecture of a battleship. Since the model is decomposed into logical units, each unit needs to maintain its own model with the relevant state information. This information is stored in a database that can be accessed by all users of the DSS system. Whenever a request is issued to the DSS system, the system accesses this database and gathers the latest state information. It uses this state to update itself before executing the simulation.

The models form a hierarchy that resembles the military command hierarchy. Whenever a user at any level of the hierarchy wishes to use the DSS system they first get the model that corresponds to the unit he is in. Next he includes as many of the models for the units below it in the hierarchy as needed to gain the appropriate resolution. These models are simply included into the simulation. Any model not included is replaced be a single distribution that models the

The System Intelligence

Each unit in the distributed architecture is to have decision making capabilities as depicted in Figure 4. While the DSS is only to suggest a COA, in the simulation it must control the system. This is the simulation must work as though it is executing the COA. The intelligent unit is actually a controller. It receives from its supervisor the tasks it is requested to process along with there associated performance functions. The performance function is a measure of the goodness of an executed COA. It may be measured in terms of monetary cost, number or personnel injured, amount of stealth etc. and is provided by the user just as the models are. Once the controller accepts a task, it decomposes the task into sub-tasks. It then negotiates the assignment of the subtasks with its subordinate controllers. It then assigns each sub-task along with the sub-task's performance function to its subordinate controllers for processing. The intelligence of the cell controller is used to find the best COA for its sub task. For example, if the goal is to move a military unit from one location to the next, the DSS may have to decide whether to go via air, ground or sea. It will run a simulation with each of the 3 methods and compare the fitness of each to determine the best. The task is decomposed by reassigning it to subordinates. For example suppose the COA to try is to go from point A to B via air and then from B to C via ground. It will give the air sub unit the task of moving the unit from point A to B. Then when the Air subordinate returns a completion notification it will assign the ground subordinate the task of moving the unit from B to C. The controllers uses simulation to generate feed-forward information that is used to predict the performance of the system under different COAs.

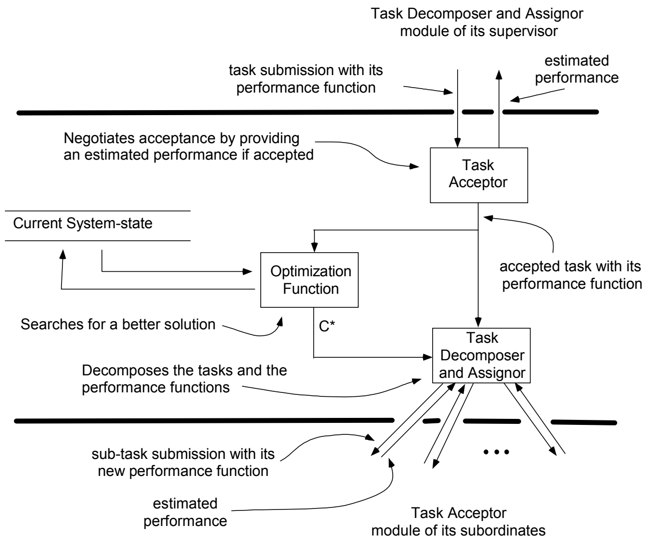


Figure 4: The three modules that make up the ICC

When the supervisor of a controller requests a task to be performed, the Task Acceptor function, see Figure 4, receives the request and negotiates its promised performance with the supervisor based on its current state. Once the Task Acceptor accepts the task, it is sent to the Task Decomposer and Assignor for execution. The Task Decomposer and Assignor then decompose the task into subtasks and assign them to its subordinate controllers. In the mean time the Optimization Function continuously searches for a better alternative COA for the specified problem and under the specified performance goals. The optimization function uses simulation to produce predicted performance information. The basic algorithm consists of running a set number of simulation runs for each alternative choice. Then using the statistical data gathered from the simulation compare which alternative performed the best. This is then the best COA. The COA may be optimal if all choices were tested or the algorithm is known to find the optimum. However since this may require more processing time than is available the choice may simply be the best choice given the amount of time and resources available to make the decision.

Conclusions

We have shown that an intelligent controller designed to control a flexible manufacturing system can be used as a decision support system in military applications. The basic decision making capabilities are valid in both applications. The distributed modelling methods that allow the control of large, physically distributed manufacturing plants become very appropriate in modelling military application. Military applications are in fact very large and distributed over a geographical area making this technology particularly useful.

While this DSS exist only in concept, it does show the benefit of exploring intelligent control research from such areas like manufacturing where much work has already been done. A military operation is like a flexible manufacturing system in the sense that they are both distributed and complex.

References

[1] Albus, J. S. and A. Meystel, "Behavior Generation in Intelligent Systems," National Institute of Standards and Technology Internal Report, 1997, Gaithersburg, MD.

[2] Albus, J. S. and A. Meystel, "A reference model architecture for design and implementation of semiotic control in large and complex systems" In Architectures for Semiotic Modeling and Situation Analysis in Large Complex Systems: Proceedings of 1995 ISIC Workshop, 33- 45, AdRem Press, Bala Cynwyd, Pennsylvania.

[3] G. Arango and R. Prieto-Diaz, "Domain analysis: concepts and research directions," Domain Analysis and Software Systems Modeling, Los Alamitos, CA: IEEE Computer Society Press, 1991, pp. 9-33.

[4] D. A. Bodner, S. A. Reveliotis, "Virtual factories: an object-oriented simulation-based framework for real-time FMS control," Proceedings of the 6th International Conference on Emerging Technologies and Factory Automation, 1997, pp. 208 –213.

[5] W. J. Davis, "A concurrent computing algorithm for real-time decision making*," ORSA Computer Science and Operation Research: New Developments in their Interfaces Conference*, Jan. 8-10, 1992, pp. 247-266.

[6] W. J. Davis, "On-Line Simulation: Need and Evolving Research Requirements," in *Simulation*

Handbook, J. Banks, ed., Wiley, 1998, pp. 465-516.

[7] W. J. Davis, J. G. Macro, A. L. Brooks, M. S. Lee, G. S. Zhou, "Developing a real-time emulation of multiresolutional control architectures for complex, discrete-event systems," *Proceedings of the Second Semiotic Modeling and Simulation Analysis Conference*, National Institute of Standards and Technology October 20-23, 1996, pp. 274-279.

[8] W. J. Davis, J. Macro, and D. Setterdahl, "An Integrated Methodology for the Modeling, Scheduling and Control of Flexible Automation," 1997, *Journal on Robotics and Intelligent Control.*

[9] W. J. Davis, D. Setterdahl, J. Macro, V. Izokaitis, and B. Bauman, "Recent Advances in the Modeling, Scheduling and Control of Flexible Automation," *Proceedings of the 1993 Winter Simulation Conference*, pp. 143-155.

[10] C. Dimopoulus, A. M. S. Zalzala, "Recent Developments in Evolutionary Computation for Manufacturing Optimization: Problems, Solutions and Comparisons," *IEEE Transactions on Evolutionary Computation*, Vol. 4, No. 2, July 2000.

[11] T. Govindaraj, L. F. McGinnis, C. M. Mitchell, D. A. Bodner, S. Narayanan and U. Sreekanth, "OOSIM: A Tool for Simulating Modern Manufacturing Systems," *Proceedings of the 1993 National Science Foundation Grantees in Design and Manufacturing Conference*, pp. 1055-1062.

[12] F. G. Gonzalez, 1996, "A Simulation-Based Controller Builder for Flexible Manufacturing Systems," *Proceedings of the 1996 Winter Simulation Conference*, pp. 1068-1075.

[13] F. G. Gonzalez, W. J. Davis, "An Intelligent Control Architecture Distributed Across The Network, For The Control Of Large-Scale Discrete-Event System," *Proceedings of the 1999 joint conference of the Third World Multiconference on System, Cybernetics and Informatics (SCI 99) and the Fifth International Conference on Information Systems Analysis and Synthesis (ISAS 99).*

[14] I. Kacem, S. Hammadi, "Approach by Localization and Multiobjective Evolutionary Optimization for Flexible Job-Shop Scheduling Problems," *Transactions on System Man and Cybernetics – Part C,* Vol. 32, No. 1 February 2002.

[15] J. H. Mize, H. C. Bhuskute, and M. Kamath, "Modeling of Integrated Manufacturing Systems," *IIE Transactions*, vol. 24, no. 3, 1992, pp.14-26.

[16] S. D. Narayanan, A. Bodner, U. Sreekanth, S. J. Dilley, T. Govindaraj, L. F. McGinnis and C. M. Mitchell, "Object-Oriented Simulation to Support Operator Decision Making in Semiconductor Manufacturing," *Proceedings of the 1992 International Conference on Systems, Man and Cybernetics*, pp. 1510-1519.

[17] W. D. Kelton, R. P. Sadowski, and D. A. Sadowski, *Simulation with ARENA*, McGraw-Hill, 2nd ed. 2001.

[18] S. S. Panwalker, W. Iskander, "A Survey of Scheduling Rules," *Operation Research*, Vol. 25, pp. 45-61, 1997.

[19] B. A. Peters, J. S. Smith, J. Curry and C. LaJimodiere, "Advanced Tutorial - Simulation Based Scheduling and Control," *Proceedings of the 1996 Winter Simulation Conference*, Eds. J. M. Charnes, D. J. Morrice, D. T. Brunner and J. J. Swain, pp. 194-198.

[20] Reyes, H. Yu, S. Lloyd, "An Evolutionary Hybrid Scheduler Based in Petri Net Structures for FMS Scheduling," *Proceedings of the 2001 International Conference on Systems, Man and Cybernetic,* pp. 2516 – 2521, V4.

[21] J. S. Smith, R. A. Wysk, D. T. Sturrock, S. E. Ramaswamy, G. D. Smith and S. B. Joshi, "Discrete Event Simulation for Shop Floor Control," *Proceedings of the 1994 Winter Simulation Conference*, pp. 962-969.

[22] J. S. Smith, B. A. Peters, "Simulation as a decision-making tool for real-time control of flexible manufacturing systems," P*roceedings of the 1998 International Conference on Robotics and Automation*, pp. 586-590.

[23] K. E. Stecke, "Design, planning Scheduling and Control problems of flexible manufacturing systems," *Annals of Operation Research*, vol. 3, 1985

Extended Generic Layered Architecture for Real-Time Modeling and Simulation

Jacek Malec[1)] and Janusz Zalewski[2)]

[1)]Dept. of Computer Science, Lund University
S-221 00 Lund, Sweden
jacek@cs.lth.se
[2)]Computer Science Program, Florida Gulf Coast University
Ft. Myers, FL 33965, USA
zalewski@fgcu.edu

Abstract

This Generic Layer Architecture (GLA) has been created to deal with problems of reasoning in conventional process control. It is composed of a process layer, dealing with the external environment, a rule layer, that provides an inference engine associated with a system state, and an analysis layer responsible for planning and reasoning. This results in a hybrid controller that enhances conventional modes of operation by the capability of taking intelligent decisions. In this work, we are extending the concept of GLA by adding into the picture three new environments, to make it suitable for purposes of modeling and simulation. The broader context includes a user interface, network communication and image database, where GLA becomes a core element of a sophisticated real-time information processing system. The extended architecture is applied to a simulation problem from air traffic control - reasoning about, predicting and detecting collisions. In this problem we focus on investigation of timing estimates on the reasoning process to meet hard deadlines in the problem domain.

Introduction

By the term *embedded systems* one usually refers to computer programs embedded in physical systems. They provide control to mechanical, chemical and other kinds of plants. The design and implementation of embedded systems usually involve the contribution from both control and computer engineers (Sanz and Zalewski, 2003).

Simple embedded systems might be designed and implemented using rather straightforward approach. The dynamics of the system in hand needs to be studied and a control algorithm may be inferred possibly using some appropriate tool, such as differential equations or a Laplace transform. An implementation of such algorithm amounts to choosing an appropriate execution model (usually periodic computations) and a computer platform, and then generating code for that platform that provably (this issue is unfortunately often omitted) implements the algorithm.

As systems become more and more complex, a plant cannot be any longer described as a set of (linear) differential equations. Firstly, such a description would take too much time to create; secondly, it would be of no use (due to complexity) for the purpose of creating a control algorithm. In such cases techniques like hierarchical description and modularization come to use.

One of the modern ways to approach the problem is to use the hybrid system models: a system is described by a family of *modes*, where each mode is described by a set of differential equations (Grossman et al., 1993). The supervisory control consists of switching among modes, depending on the state of the plant.

Such systems are called *hybrid*, since they consist of continuous subsystems and a discrete supervisory structure. To implement hybrid controllers one needs to provide the algorithms for each control mode and the discrete controller to perform the mode switching and handle the interaction between the continuous and the discrete control.

Implementing hybrid controllers using languages like C or Ada is rather cumbersome since they offer very poor support for algorithms expressed using state machines. Software support and computer aided tools are therefore necessary to assist in the design of hybrid controllers. One such tool is the Generic Layered Architecture, GLA, (Morin et al., 1992) for implementing hybrid controllers in software. The languages and tools developed for the architecture support both periodic and discrete computations required by hybrid controllers.

However, the software development tools are only a part of a larger picture - first the software needs to be specified and designed. Even better if this specification could be formal, so that validation and verification techniques could be used with their full potential, yielding provably correct embedded system. But, in order to get confidence in formal tools, one needs a faithful model of a (hybrid) controller and its environment.

The GLA has been previously used as a modeling tool (Malec et al., 1995), however at that time no verification support tools have been incorporated in the GLA methodology. Due to increasing complexity of considered applications, but also due to the availability of the more powerful computer technology, the next step towards a more powerful modeling and analysis methodology could be taken.

In this paper we describe the extended GLA (eGLA) obtained by adding into the picture three new environments, to make it more suitable for purposes of modeling and simulation. The broader context follows the fundamental principles of designing real-time software architectures (Zalewski, 2001) and includes a user interface, network communication and image database, where GLA becomes a core element of a sophisticated real-time information processing system. The extended architecture is applied to a simulation problem from air traffic control - reasoning about, predicting and detecting collisions. In this problem we focus on investigation of timing estimates on the reasoning process to meet hard deadlines in the problem domain.

The paper is divided as follows. First, the original GLA is introduced, together with the associated software and formal analysis tools. Then the extensions to the architecture are described. Finally, we present the problem from air-traffic control and sketch the advantages of using eGLA in this context.

The Generic Layered Architecture

Our approach to hybrid system design and implementation builds on the three-layered software architecture GLA, the generic layered software architecture, developed since the early nineties (Morin et al., 1992). The architecture differs from other layered approaches by grouping similar types of computations into *layers* (shown in Figure 1), as opposed to functional decomposition.

The first layer, called the *process layer* (PL), is intended to host implementations of numerical, periodic tasks, such as identification or control. Data handled by this layer are contained in input and output vectors. Computations have the form of mappings from input vectors to output vectors and are performed periodically in synchronization with the sample rates of sensors.

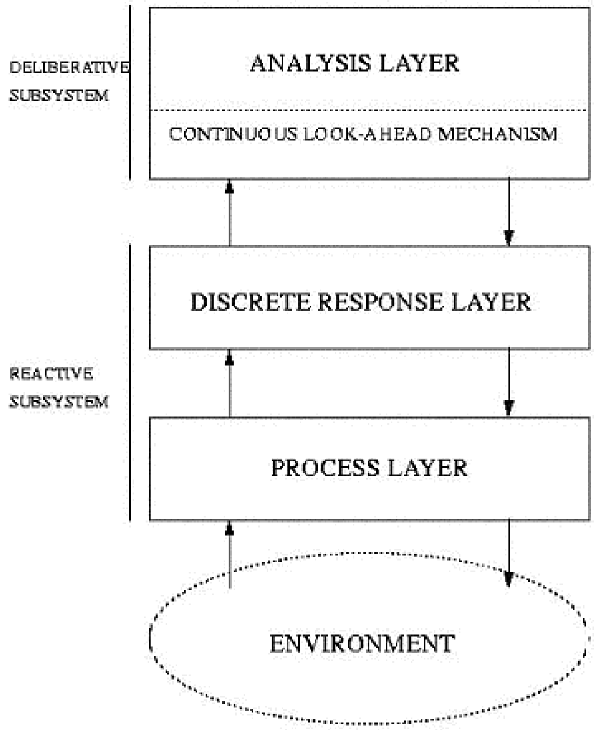


Fig. 1. The generic layered software architecture.

The middle layer is called the *discrete response layer* (DRL), and performs tasks, which are by nature asynchronous. For instance, it computes the response to asynchronous events that are recognized in the PL. An example of such a task is the change of control mode (of the PL) due to the change of mode in the environment. The computational model assumed for the DRL is that of discrete event systems (DES). There exist several equivalent DES formalisms: automata, transition systems, rule-based systems, etc. All of them distinguish the notions of *state* and *transition* as central, although the details vary from model to model. We have used the rule-based approach for the purpose of specifying knowledge-based system prototypes. However, this does not preclude usage of other approaches (Malec, 1992).

The top layer is called the *analysis layer*, because it is intended to handle symbolic reasoning tasks such as prediction, planning and scheduling, which require reference to physical time. The output of this layer can be either control events that guide the DRL in its decisions or parameter settings that are passed through the DRL and directly affect the operation of the PL.

Real-Time Software Tools

This architecture and its implications on software engineering issues have been thoroughly studied in previous research (Morin et al., 1992). One of the conclusions was that it facilitates prototyping of systems, especially because it allows development of generic software tools, which can be used for implementation of each particular application system. Along this line software kernels, or *engines*, have been developed for construction and implementation of the process layer and the discrete response layer. The set of tools includes:

- Process Layer executive, PLX (Morin, 1993): a multi-threaded time-triggered real-time engine for implementation of process layer software. It has been implemented on a pSOS based system, on a PC running VDX, and recently on a RTLinux-based PC. There also exists a simulator of the PLX, which runs under Unix;
- Process Layer Configuration Language, PLCL, (Morin, 1991) and its compiler: a language for specification of PL module interconnections and interfaces to both sensors and actuators on one side, and to the DRL software on the other side. The modules themselves are programmed in a subset of some conventional language, such as C or C++;
- Rule Layer executive, RLX (Morin, 1994): an engine for implementation of rule-based discrete-event systems in the DRL. It has been implemented on Unix-based machines, and on a PC with VDX and RTLinux;
- Rule Language. RL (Morin, 1994): a rule-based language for declarative specification of discrete-event control.

The Process Layer Executive

The PLX supports the implementation and maintenance of the hard real-time parts of the application, which perform the transformations of periodic data. During processing, all data are stored in a *dual state vector*, which is a global data structure consisting of an input and an output vector. The values in the input vector represent either sensor readings or internal state, whereas the values in the output vector represent actuator outputs or new internal state.

A PL application defines a sequence of transformations that should be applied to the input vector in order to compute a new output vector. Since sensor values are read periodically, the transformations have to be applied with the same periodicity. When the transformations have been applied, actuator outputs are flushed to devices and the new internal state is installed in the input vector, thus establishing the new state of the process layer.

The PLX engine supervises a PL application, which includes managing the internal state, reading sensors and writing to actuators using user defined access functions, and supervise the execution of the periodical transformations of the dual state vector. The PLX supports decomposition of the vector into several sub-vectors, each of which has its own period, causing transformations to be executed at different rates.

The PLX engine and the PLCL compiler together form the basis for a worst-case execution timing analysis of a PL application (Morin, 1993).

The Rule Layer Executive

The rule layer executive supports the implementation of rule-based event processing. In the RLX the state of the world is represented by the time dependent values of a set of symbolic state variables called *slots*. A slot is updated only when its value changes, due to an external event or as a result of a change of another slot's value. Rules specify dependencies between slots, and typically have the following form: *if the value of a particular slot is changed in a certain way, then the value of another slot should also be changed as a result*. Internally, each such change may trigger additional rules, which lead to more updates. This forward chaining process may continue in several steps.

We take an object-oriented view of the rule base in the sense that we associate a set of rules with each slot. This view facilitates the flexibility and maintainability needed for complex systems.

The RLX tool has two major tasks. Firstly, it maintains the set of slots and rules, which determines the behavior of the discrete response layer. Secondly, it performs the forward chaining of rules. The forward chaining is typically initiated by an event recognized by the PL. The result of the forward chaining process may be the change of control algorithm used by some PL process or direct output to a device interfaced by the PL.

The RL is essentially a syntactic variant of a simple temporal logic and is used for declarative specification of behavior of the discrete response layer. A comprehensive set of tools for correctness and consistency checking, for timing analysis of RL programs and for code generation have been developed during the recent years (Lin and Malec, 1998; Lin, 1999a; Lin, 1999b).

The Extended GLA

The GLA serves well as a medium for embedded controller implementation, but its use as a tool for system modeling and analysis, although foreseen from the very beginning (Malec et al., 1995), is somehow limited. This is mainly due to the lack of support tools that would help the engineer to formulate the problem in a language of their choice, create a simulation for deeper understanding and finally perform analysis using formal and informal tools available. In addition to that, the model of the controller can be extended to include not only the process aspects currently implemented in the Process Layer, but also three additional aspects of an architecture a of modern real-time controller, as illustrated in Figure 2 (Zalewski, 2001):

- Graphical User Interface, GUI, allowing the operator to interact in real time with the controlled process and other components of the control system;
- Network Communication, handling the connectivity among multiple processors or computing nodes forming the control system; and
- Database Interface, allowing for real-time access to permanently stored data, such as images in real-time simulation, flight plans in air-traffic control, etc.

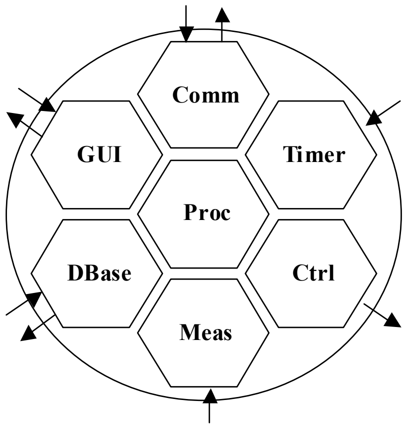


Fig. 2. Architectural template for real-time software design

Assuming that the GLA's Process Layer is represented in Figure 2 by the Measurement and Control components, and the upper GLA layers are embedded in the Processing component, the extensions proposed just add independent run-time agents, similar to the PL, concentrating on three orthogonal I/O functions. Therefore the extension should not be treated as additional layers of the architecture. The division of GLA into three layers, based on the computational pattern employed, still holds.

However, some functionality should be available to the end-users if the tool is to fulfil its functions. Thus the extensions should be seen as separate functionalities available at the lowest layer. All three extensions, User Interface, Network Connectivity, and Image Database, are described briefly below.

User Interface

No software, whether it's the controller itself or the tool to design it, can expect popularity without an intuitive, easy-to-use GUI. In an early phase of language design, there are usually no additional tools, but before such development can be considered mature there needs to be a number of user interfaces allowing a non-expert user to exploit all the advantages of the tool without making unnecessary mistakes.

GLA was lacking the user interface for years, until very recently, when the design and implementation have begun. The currently developed interface allows the user to develop PL programs and test their timing properties, to develop RL programs and run various semantic and timing checks, to develop lookahead reasoning in the analysis layer and test its connection with the lower layers of the software. There exist also tools allowing import of specifications done using external formalisms, like e.g., the hybrid automata developed using HyTech tool. Moreover, a stepping and debugging facility exist for all three layers of the software.

At the tool level, we consider introduction of visualisation mechanisms for better illustrating the interplay between the discrete and the continuous part of the software, between the system and its environment (including other embedded software developed using the GLA approach, see below) and of the analysis layer software.

Network Connectivity

In order to use a GLA-based system in a larger setting, including other computer-based control or, especially, simulation systems, one needs to provide connectivity to the outer world. The natural choice for this is TCP/IP networking allowing pieces of software running in different systems to communicate with each other according to predefined protocols.

In general, all three levels of the GLA architecture can communicate smoothly over the network with other networked applications, possibly but not necessarily also GLA-based. However, in this project we are restricting the network connectivity to the lowest layer agent responsible exclusively for maintaining data communication. This allows distributed algorithms, such as large-scale simulations, to be easily developed and run. The applications described below provide a good example of advantages of distributed simulation and power of analysis tools available for the GLA.

Image Database

Current state-of-the-art simulation and analysis tools require access to large databases containing data in various formats. A particularly challenging topic is the use of large corpora of image data available. We expect that applications developed using GLA might exploit such databases and therefore the third extension of GLA is the mechanism of accessing them.

Usually such need arises at the level of discrete model, when a control mode is to be changed (e.g., classification of an observed plane as a civilian line aircraft rather than a military UAV, or classifying a ship as a ferry carrying people rather than a cargo ship), but the implementation of the mode change will usually rely on computations performed on the lowest level. Therefore the main interface will be implemented at the process layer level rather than any of the other two layers.

Case Studies

Benchmarking and Training System Simulation

To verify the concept and evaluate timing properties of eGLA, several experiments were conducted within the experimental testbed composed of three high-level software design tools imitating the behavior of the Process Layer (VxWorks) and User Interface (ObjecTime and Rose Realtime), with SES/workbench acting as a Discrete Response Layer (Mathure at al., 2003). First, the tools were mapped onto a five-task benchmark as described in (Guo et al., 2003) and illustrated in Figure 3.

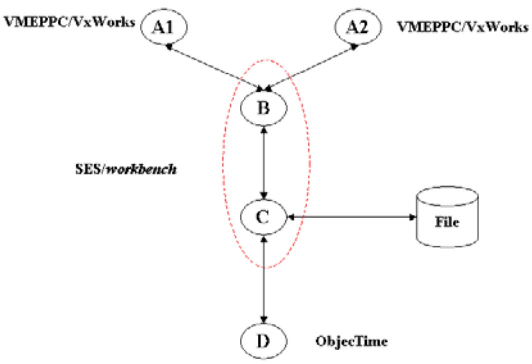


Fig. 3. Five-task benchmark simulation architecture.

The benchmark is consistent with the architectural template presented in Figure 2 and is composed of the following components:

- two measurement processes, A1 and A2, representing the Process Layer, run on VxWorks real-time kernel
- a computational process, B, representing the two upper layers of the GLA, run within SES/workbench
- a database process, C, representing a database interface, run also within SES/workbench, and
- a GUI process, D, run within ObjecTime.

The idea of this experiment was to evaluate performance of the model of an extended GLA, in a heterogeneous and complex environment, where multiple tools based on different theories (differential equations, finite state machine, and queuing theory), but integrated via TCP/IP protocol suite, contribute to the overall goal. The experiments proved that the integration is possible and stable operation is achieved within a short period of time.

The next step was to use the testbed for a more sophisticated case study consisting of a training system developed for combat vehicles (Al-Daraiseh et al., 2000). A general configuration of this system consists of the following four building blocks:

- SMC, Simulation Management and Control, which is a central part of the simulation, running within SES/workbench
- ITS, Intelligent Tutor and After Action Reporting, which runs under VxWorks
- IG, Image Generation Subsystem, also running under VxWorks
- CGF, Computer Generated Forces, simulated in ObjecTime/Rose Realtime.

The performance of this system was evaluated by simulating SMC module as a server in the role of the GLA's DRL layer. It served 9 clients delivering nine types of periodic messages and one additional client generating high-priority non-periodic traffic, according to the customer's specification (Al-Daraiseh et al., 2000). The impact of changing arrival rates on queue length was analyzed for both types of traffic. As shown in Fig. 4, sample measurements for periodic traffic only and three different distributions of the server population. The results show faster system saturation for longer service times, as expected (Mathure et al., 2003).

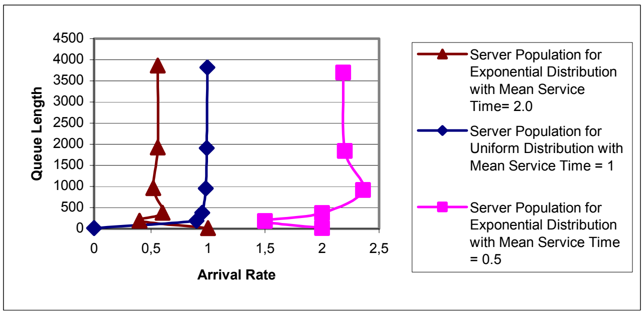


Fig. 4. Queue length vs. arrival rate for periodic and non-periodic traffic combined.

Air Traffic Conflict Detection and Resolution

The objective of this case study is to focus on the capability of the DRL layer in the extended environment and apply the Extended GLA to a novel algorithm suite for airspace management and deconfliction of multiple Unmanned Aerial Vehicles (UAV's). To avoid collisions in space, ideally all vehicles should have enough information of all other vehicles' positions, directions and speeds. This idea, however, is not feasible and effective to implement, since it is only good under the assumption that vehicles are cooperating and a computational unit in each vehicle has enough memory and processing power (speed) to complete computations on time, before a collision occurs. For example, for N vehicles, assuming each vehicle has to compute its X parameters for the next time period to predict collisions, the computational time is proportional to (N-1)*X, which is linear with respect to the number of vehicles, but has two disadvantages:

- grows significantly with a number of vehicles involved and may reach limits of processing capabilities, and
- is unpredictable in highly crowded environments, because there is no way of knowing in advance how big N can become.

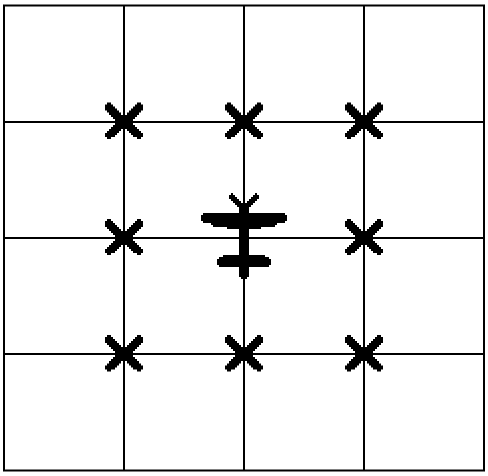


Fig. 5. UAV and its neighbor zones in 2D space.

Thus, a collision avoidance method is needed, which can offer reduction in computational space and time, additionally ensuring collision predictability under all circumstances. Initial reduction of computational complexity can be achieved by reducing the problem to a two-dimensional space and assuming that only immediate neighbors can participate in a collision, as illustrated in Figure 2. Then, the computational time reduces and is never greater than 8*X, which means that its upper limit does not depend on the number of vehicles involved.

A new method called Right-Of-Way (ROW) deconfliction further reduces the computational time to 3*X, because only three immediate right-hand neighbors are involved in computations, assuming that all vehicles in space observe the Right-Of-Way principle and use the same algorithm of deconfliction.

A block diagram of software architecture of ROW deconfliction is shown in Figure 6. The reasoning system consists of two major parts: Trajectory Prediction module and Conflict Detection module, both forming the DRL layer in eGLA. The Trajectory Prediction module takes inputs from its own vehicle and obtains global time. It also responds to inputs from Command and Control (equivalent of GUI). UAV inputs come from vehicle sensors. Based on this information the module estimates and predicts values of state of the vehicle, using Kalman filter as a computational model. The results of computations are periodically fed to the Collision Detection module, which combines it with information obtained from participating neighbor vehicles and with Command and Control to use it as a basis for decision making.

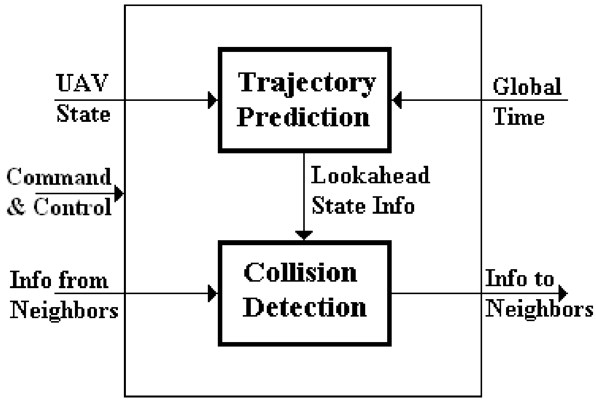


Fig. 6. Software architecture of the ROW system.

The model of the dynamics assumes a two-dimensional linear system described using a set of general state-space equations of the form (Johnson et al., 2003):

$$\mathbf{X}_{k+1} = \varphi_k \mathbf{X}_k + \mathbf{w}_k$$
$$\mathbf{Z}_k = \mathbf{H}_k \mathbf{X}_k + \mathbf{v}_k$$

where $\mathbf{X}_k$ is an (n×1) process state vector at time t_k, φ_k is (n×n) state transition matrix, $\mathbf{w}_k$ is (n×1) vector, assumed to be a white sequence with a known covariance structure, $\mathbf{Z}_k$ is (m×1) measurement (observation) vector at time t_k, $\mathbf{H}_k$ is (m×n) matrix giving the ideal (noiseless) connection between the measurement and the state vector at time t_k, and $\mathbf{v}_k$ is (m×1) measurement error, assumed to be a white sequence with known covariance structure and having zero cross-correlation with the $\mathbf{w}_k$ sequence.

If this two-dimensional model works properly for surface based UAV applications, such as airport taxing, factory floor or parking lot, the three-dimensional model will be built in the next step to resolve conflicts in air space.

Conclusion

The presented research is the first step towards making the concept of Extended GLA a useful tool for modeling and simulation, as opposed to pure implementation of hybrid intelligent control systems. Thus far, extending GLA by adding three more orthogonal components at the Process Layer level turned out to be useful for specific applications considered, as confirmed by simulation experiments.

Further work is needed on extending GLA toolset to support the design process with appropriate user interfaces and network and database connectivity. This will significantly enhance the development process currently supported mostly by off-the-shelf tools.

Acknowledgements

Some of the ideas described in this paper have been developed within master-level projects at the Department of Computer Science of Lund University (GLA), and at the Computer Engineering Program of the University of Central Florida (real-time software architecture).

References

Al-Daraiseh, A., Al Mazid, A., Croeze, T., Zalewski, J., Dolezal, M., Green, G., Bahr, H. (2000). High-Level Tool Support for Integration Architecture in a Distributed Embedded Simulation Project, *Proc. CSMA2000, 2nd Conference on Simulation Methods and Applications*, Orlando, Florida, October 27-29, pp. 33-36

Grossman, R.L., Nerode, A., Ravn, A.P., & Rischel, H., editors. (1993). *Hybrid Systems*, Berlin: Springer-Verlag.

Guo, D., van Katwijk J., Zalewski, J. (2003). A new benchmark for distributed real-time systems: Some experimental results, *Proc. WRTP'03, 27th IFAC/IFIP Workshop on Real-Time Programming*, Łagów, Poland, May 14-17.

Johnson, R., Sasiadek, J., Zalewski, J. (2003). Kalman Filter Enhancement for UAV Navigation, *Proc. SCS 2003 Collaborative Technologies Symposium*, Orlando, Fla., January 19-23, pp. 262-272.

Lin, M. (1999a). *Analysis and Synthesis of Reactive Systems: A Generic Layered Architecture Perspective*, PhD thesis, Department of Computer Science, Linköping University.

Lin, M. (1999b). Timing analysis of {PL} programs. *Proc. WRTP'99, 24th IFAC/IFIP Workshop on Real-Time Programming,* Dagstuhl, Germany, May 30 – June 3.

Lin, M., & Malec, J. (1998). Timing analysis of RL programs. *Control Engineering Practice*, *6*, 403-408.

Malec, J. (1992). Complex behavior specification for autonomous systems. *Proc. IEEE International Symposium on Intelligent Control'92*, pp. 178-183, Glasgow, Scotland,.

Malec, J., Morin, M., & Nadjm-Tehrani, S. (1995). A layered software architecture for design and analysis of embedded systems. *Proc. 1995 International Symposium and Workshop on Systems Engineering of Computer Based Systems*, pp. 169-176, Tucson, Ariz.

Mathure M., Jonnalagadda V., & Zalewski J. (2003). Heterogeneous architecture and testbed for simulation of large-scale real-time systems. *Proc. 7th IEEE Int'l Symp. on Distributed Simulation and Real-Time Applications*, pp. 37-42, Delft, The Netherlands.

Morin, M. (1991). *PLCL - Process Layer Configuration Language.* Technical Report LAIC-IDA-91-TR10, Linköping University.

Morin, M. (1993). *Predictable cyclic computations in autonomous systems: A computational model and implementation*. Licentiate thesis 352, Dept. of Computer and Information Sciences, Linköping University.

Morin, M. (1994). *RL: An embedded rule-based system.* Technical Report LAIC-IDA-94-TR2, Linköping University.

Morin, M., Nadjm-Tehrani, S., {\"O}sterling P., & Sandewall E. (1992). Real-time hierarchical control. *IEEE Software*, *9*(5), 51-57.

Sanz, R. & Zalewski J. (2003). Pattern-Based Control Systems Engineering, *IEEE Control Systems*, *23*(3), 43-60.

Zalewski J. (2001). Real-time software architectures and design patterns: Fundamental concepts and their consequences, *Annual Reviews in Control*, *25*(1), 133-146.

Medical Simulation

The Effectiveness of Training Mental Rotation and Laparoscopic Surgical Skills in a Virtual Environment

Harshal Sangani[2], Kristy A. Bradley[1], & Cali Fidopiastis[2,3]

[1]University of Central Florida
[2]Institute for Simulation and Training
3280 Progress Drive,
Orlando, FL 32826
[3]School of Optics/CREOL
4000 Central Florida Blvd.
Orlando, Florida 32816-2700
hsangani@ist.ucf.edu

Abstract

Current research in the area of medical simulation is investigating the efficacy of using virtual reality surgical simulators as a means of training surgical skills. More recently, medical simulation has focused on the use of virtual reality to train laparoscopic skills. Several traditional methods of training laparoscopic skills exist (e.g., pelvic trainers, video trainers, animal trainers). However, the medical community is now shifting its attention toward higher fidelity simulations and virtual reality in particular for a number of reasons. Simulations provide a quicker, more cost-effective method for training and practicing surgical skills. In addition, simulators minimize the amount of physical space required for training. Nonetheless, it has been reported that the transfer of training rate from a medical simulator was only approximately 38%. Therefore, there has been a movement to refine medical simulations by focusing on improving design and haptic technology. Current simulators utilize high-level graphics to more realistically simulate deformable tissues in order to improve visualization of organ properties. In addition, they employ force feedback in order to give the simulation a more genuine haptic performance. Both visualization of organ properties and realistic haptic feedback are characteristics lacking in traditional mechanical trainers.

Many experimental comparisons of traditional training methods to medical simulators, including laparoscopic simulators, exist in the current literature though few account for spatial ability when evaluating successful skills training. Spatial ability has been shown to significantly affect training in virtual environments. More specifically, inferior mental rotation skills are correlated with poorer performance on surgical tasks in real world and virtual environments. As such, the current study investigated the effects of mental rotation skills on the ability to acquire and perform laparoscopic suturing skills. In addition, a subsequent study is proposed that will experimentally compare laparoscopic surgical skill levels attained from training with the Reachin Laparoscopic Trainer to those obtained from a traditional mechanical pelvic trainer. Prior to training on the simulators, participants will be evaluated on their mental rotation ability and subsequently trained to commensurate levels using an existing mental rotation training paradigm.

Introduction

Although abdominal cavity surgery has been practiced for the last 100 years, more advanced procedures such as cholecyctectomy (gallstone removal) were only performed in last 20 years (Himal, 2002). Cholecyctectomy is an example of laparoscopic surgery, which in turn falls into the category of endoscopic or minimally invasive abdominal surgical procedures. Minimally invasive surgery has the advantage of reducing damage to healthy tissue, recovery time, and infection rate. These advantages for the patient come as a challenge to the surgeon learning these new techniques.

During a laparoscopic surgery, the surgeon loses direct contact with the operation site. In the current procedure an endoscope, or tubing attached to an optical system, projects video images of the internal organs and tissue into a monocular viewing display. The surgeon manipulates surgical tools for suturing and tissue manipulation with a loss of visual feedback and tactile stimulation (Szekely, Haller, & Bajka, 2000). A current review of British and Australian laparoscopic procedures suggests that these surgeries are more difficult, time consuming, and harder to learn (Molloy, 2001). Molloy suggest that not all surgeons possess the visuo-spatial skills necessary to perform laparoscopic surgery. Most literature reviewed cited limited and ineffective training paradigms as the number one problem for surgeons performing these types of surgeries.

Typically, doctors learning endoscopic procedures attend a weekend seminar to learn the methods, and then practice using animals such as pigs (Durlach & Mavor, 1995). In some cases, this training does not allow the doctor to reach proficiency before proceeding to human patients. Mechanical simulators were introduced in the early 1990's (Satava, 1998). Although still in use, these simulators are rigid and biologically unrealistic. Surgeons performing laparoscopy suffer from restricted vision and lack tactile perception. Current simulators, for example KISMET, utilize high-level graphics to more realistically simulate deformable tissues to improve visualization of organ properties. In addition, they use a "force feedback probe to give the simulation a more genuine haptic performance. However, problems of fidelity and realism still exist in designing medical trainers. Of more concern is ability to train surgical dexterity and decision-making. These skills might rely more on the surgeon's ability to mentally rotate objects and their comfort in pressure scenarios (Wanzel, Hamstra, Anastakis, Matsumoto, & Cusimano, 2002).

Though many researchers have sought to compare or evaluate the effectiveness of various types of laparoscopic trainers to train laparoscopic skills, few have controlled for the individual's ability to mentally rotate objects, such as organs. Tendick (2003) notes that high correlations have been observed between standardized spatial ability tests and performance ratings on open surgery tasks. However, Tendick proposes that laparoscopic surgery might be even more dependent on spatial ability due to the lack of available perceptual information leaving surgical actions dependent on internal representations of surgical information. Eyal and Tendick (2001) investigated the effects of spatial ability on learning the use of an angled laparoscope in a virtual environment and found support for their hypothesis that those who score higher on standardized spatial ability tests will perform better in the laparoscopic simulation. Related to spatial ability and its effects on laparoscopic skills is the gender of the surgeon performing the surgery.

Much of the research conducted to date has demonstrated a gender difference in spatial ability such that males perform at a higher level than females (Rizzo, Buckwalter, Neumann, Kesselmann, Thiebaux, Larson, & van Rooyen 1998; Peters, Laeng, Lathan, Jackson, Zaiouna, & Richardson, 1995; Peters, Chisholm, & Laeng, 1995; Masters & Sanders, 1993) The current study seeks to replicate the gender difference findings of Rizzo et al regarding gender and spatial ability. In addition, the current study will evaluate the effects of mental rotation skills on the ability to acquire and perform laparoscopic suturing skills. In congruence with the previous findings in the literature, it is expected that females in both the control and experimental groups will perform worse in the paper and pencil version of a mental rotation test. Furthermore, it is expected that these gender differences will be eliminated after participants receive training with the virtual reality mental rotation task. With regard to performance in the surgical environment, it is expected that those participants who received mental rotation training will perform better on the suturing task exams.

Method

Participants

Twelve university students (6 males and 6 females, mean age = 22.67 years) were recruited to participate in the experiment. All participants were right handed.

Materials

To test initial mental rotation ability, The Mental Rotations Test (Peters et al., 1995), a paper-based mental rotation test was used. In addition, the University of Mississippi Mental Rotation Experiment was presented online. A computer based mental rotation task was constructed using 3D objects generated in 3D Studio Max. The 3D computer based task was used for the object rotation pre and posttest as well as for the training phase.

Procedure

Participants completed the informed consent form and demographics questionnaire upon arrival. Participants were then asked to complete the paper-based mental rotation test and the online mental rotation test. The presentation of these tests was counterbalanced to prevent order effects. Following these tests, the participants completed a 20 item pre-test in which they were to rotate a 3D object in 3D Studio Max to match a static 3D object (also generated in 3D Studio Max) presented in Microsoft PowerPoint on an adjacent computer. The objects generated were comparable to those found in the standardized paper-based mental rotation test. Those participants in the experimental condition followed the pre-test with 10 blocks of 10 trials each of the object rotation in 3D Studio Max. For this experimental phase, 10 screenshots of each of 10 original 3D stimuli were taken. Though each stimulus type was presented in the same order to each participant, the 10 static screenshots of each were presented in a randomized order for 30 seconds on the adjacent computer in PowerPoint. Participants were again required to rotate the original object in 3D Studio Max to match the object presented in PowerPoint. The experimental group then completed the same items on the 20 item pre test in the form of a post test. The total time required to rotate and orient the object for each trial of the pre, experimental, and post test was measured in seconds via a stopwatch.

After the mental rotation portion of the experiment, all participants were provided with suturing training by an endoscopic surgeon. More specifically, participants were shown the procedure for completing the endoloop suturing procedure, the extracorporeal curved needle procedure, then the intracorporeal knot-tying procedure. Participants were trained in accordance with the typical procedure the surgeon utilizes for student training. For each procedure, the participants were instructed orally then watched the surgeon perform the suture in a standard box trainer with a 2D monitor. Following the instruction by the surgeon, the participants practiced the procedure one time. After the instructional and practice sessions, each participant was required to perform the 3 procedures while being evaluated by one of 2 experienced surgeons. The participants were rated on technique and the time to complete each procedure was recorded.

Results

The demographic questionnaire revealed that participants represented several disciplines of educational study: 3 Engineering, 4 Computer Science, 1 Biology, and 4 Psychology. Only one participant had prior experience internal surgery, while four had experience with virtual environments. Over half of the participants played video games (67%). The average of video game play was 3.5 hours, SD = 6.07.

For the paper and pencil test, a significant difference was found between the females ($M = 7.0$ $SD = 0$) and males ($M = 14.67$, $SD = 2.52$) of the control group only, $t(4) = -5.77$, $p < .01$, a = .05. While the females of the experimental group outscored the males ($M = 13.33$, $SD = 5.03$ and $M = 9.33$, $SD = 7.09$, respectively), this difference was not significant.

Although the total time to perform the online mental rotation task was highly correlated with same or different stimuli, $r(6) = .988$, $p < .001$), the time was not correlated with scores on the paper and pencil version of the test. However, number of hours spent playing video games was positively correlated with the paper and pencil test scores, $r(6) = .818$, $p < .05$).

Neither gender showed significant differences in time (seconds) to rotate the 3D object after training. However, males ($M = 11.23$, $SD = .86$) performed better on the posttest after training than females ($M = 13.38$, $SD = 3.16$). These differences were not statistically significant.

One male participant from the experimental group could not participate in the surgical training aspect of the experiment. Although this data was analyzed for total time for each surgery and number of errors, there was not enough statistical power to show differences between the genders or the two testing conditions.

Discussion

The small sample size limited the conclusions of this study. However, there were two important outcomes that cannot be disregarded. First, females in the experimental group outscored the males by 4 points. Given the scoring procedure used for the Mental Rotation test, females in the experimental group answered 4 more questions correctly than their male counterparts.

Although the sample size was small, the fact that all three of the experimental group females played video games while none of the control group females did seems to suggest that females who play certain video games may achieve the same or higher metal rotation abilities than males. Moreover, only one male out of the 6 total did not play video games. Thus, females in the experimental group were possibly better matched to the males in whole sample population. This conclusion is supported by the positive correlation between female scores on the mental rotation test and number of hours playing video games. Given the unequal match between video game players and gender, we cannot replicate the results of Rizzo et al. until we attain a more balanced sample.

In addition, the initial hypothesis that females would perform worse than males in both the experimental and control group was not shown. The mental rotation results possibly suggest that the gender differences seen in our small sample are ones that can be attributed to strategy of performing the mental rotation task. Saucier et al. (2002) have shown that women use more landmarks during way finding task, while men use Euclidean measures. The Euclidean approach is highly correlated with scores on mental rotation tests while landmark use is not. Further investigation of video game use and mental rotation abilities are planned for future research using the virtual reality mental rotation task.

A second important result is that all naïve participants learned all three surgical procedures in less that an hour. The mean total times (seconds) did not show an effect of mental rotation training, (M = 476.14, SD = 111 for females, and M = 412.83, SD= 87.45 for males). This result could be due to the confound of video game play. This conclusion is not supported in the literature. Another explanation could be that these surgical procedures relied more on depth perception than mental rotation. A future study will determine a more appropriate surgical task that utilizes mental rotation, such as determining the location of vital landmark before suturing.

Although the hypotheses were not supported for the surgical aspect of the experiment, the results do support that novice participants with no prior surgical training are adequate for studying this type of research question. Moreover, the participants were rated quite high on their surgical performance scores. Despite confounds of the small sample size and unmatched groups, the experimental paradigm is valid and can produce reliable results when implementing the full design.

The knowledge gained from the pilot study will help to guide the next phase of this experiment, which is to compare the Reachin virtual laparoscopic trainer to the pelvic box trainer. A sample size of at least 48 subjects is needed for this study. In addition, a second study on the effects of video game use, gender, and mental rotation is planned to eliminate video game experience as a confound in the experimental design.

References

Durlach, N. I., & Mavor, A. S. (1995). *Virtual Reality: Scientific and Technological Challenges*, Washington, DC: National Academy Press.

Eyal, R., & Tendick, F. (2001). Spatial ability and learning the use of an angled laparoscope in a virtual environment. In J.D. Westwood, H.M. Hoffman , G.T. Mogel, & D.Stredney (Eds.), *Machine Meets Virtual Reality* (pp. 146-152). Amsterdam: IOS Press.

Himal, H.S. (2002). Minimally invasive (laparoscopic) surgery. *Surgical Endoscopy*, *16*, 1647-1652.

Masters, M. S., & Sanders, B. (1993). Is the gender difference in mental rotation disappearing? *Behavioral Genetics, 23*, 337-341.

Mental rotation experiment (n.d.). Retrieved from the University of Mississippi, Department of Psychology Website: http://psychexps.olemiss.edu/Exps/Mental_ Rotation /startmr.htm

Molloy D. (2001). Solutions for the born-again evangelists. *Gynecological Endoscopy*, *10*, 1-5.

Peters, M., Chisolm, P., & Laeng, B. (1995). Spatial ability, student gender and academic performance. *Journal of Engineering Education, 84*, 69-73.

Peters, M., Laeng, B., Lathan, K., Jackson, M., Zaiouna, R., & Richardson, C. (1995). A redrawn Vandenberg and Kuse mental rotations test: Different versions and factors that affect performance. *Brain and Cognition, 28,* 39-58.

Rizzo, A.A., Buckwalter, J.G., Neumann, E., Kesselmann, C., Thiebaux, M., Larson, P., and van Rooyen, A. (1998). *The virtual reality mental rotation spatial skills project.* CyberPsychology and Behavior, 1*(2), 107-113.*

Saucier, D.M, Green, S. M., Leason, J, MacFadden, A., Bell, S., Elias, L. J. (2002). Behavioral Neuroscience, *116(*3), 403-410.

Satava, R. M. (1998). *Cybersurgey: advanced technologies for surgical practice.* New York: John Wiley & Sons.

Szekely G., Haller U., Bajka M. (2000). Virtual reality-based endoscopic surgery. *Presence*, *9*(3), 310-333.

Wanzel K.R., Hamstra S.J., Anastakis D.J., Matsumoto E.D., Cusimano M.D (2002). Effect of visual-spatial ability on learning of spatially-complex surgical skills. *The Lancelt*, *359*, 230-231.

Developing and Implementing a Human Patient Simulation Curriculum for Junior Medical School Students

Valeriy V. Kozmenko, MD
Kraig S. DeLanzac, MD
Jason A. Rigol, MD
Sheila W. Chauvin, Ph D
Tong Yang, MD, MS
Charles W. Hilton, MD

Anesthesiology and Intensive Care Simulation Lab
LSU School of Medicine – New Orleans

2020 Gravier St. Suite 102
New Orleans, LA 70112
vkozme@lsuhsc.edu

Abstract

We used the commercially available Human Patient Simulator – Version 6.1 (METI, Inc.) to design and implement what we believe to be one of the first curricula employing simulation for U.S. medical students. Each junior student undergoes nine sessions with the simulator learning to manage complex emergency medical problems before encountering them in the real world. This program covers simulation of the following conditions: acute thermal injury, acute upper GI bleeding, tension pneumothorax, CHF, atrial fibrillation, acute asthma attack with severe bronchial obstruction, septic shock, ARDS, and eclampsia. One session is used as an introduction to the simulation system, and the other sessions are equally spread among four major clerkships: medicine, obstetrics/gynecology, pediatrics, and surgery. Effectiveness of the sessions is assessed by administration of pre-activity and post-activity quizzes, and by grading student performance during the simulation sessions. We believe that we have successfully linked the content and learning objectives of the simulation sessions to the knowledge level of our junior medical students. The use of the simulator enhances effective retrieval and application of clinically important knowledge of complex medical conditions and facilitates the development of practical skills in patient management. This paper will present an overview of the approach that the LSU School of Medicine – New Orleans has taken in the development of this curriculum.

Introduction

Human patient simulation has been actively used in formal medical education in the medical academic world. Although the effectiveness of simulation has been appreciated subjectively by many scholars and has been confirmed objectively by several studies, at the present time human patient simulation has not been widely incorporated into medical school curricula. The Louisiana State University School of Medicine – New Orleans has developed and implemented an innovative, year-long curriculum for junior medical students using a high fidelity Human Patient Simulator (HPS) (METI, Inc.). Every third year medical student participates in eight simulation sessions equally spread among four major clerkships. In developing and implementing the simulation curriculum, there were several important issues to address:

1. organizational preparation
2. scenario content
3. learning objectives
4. session administration
5. efficacy measurement

Organizational Preparation

To effectively budget the time of the Anesthesiology and Intensive Care Simulation Lab we needed to determine:

1. number of clerkships that would send us students
2. number of simulation sessions that could be effectively delivered per day
3. number of students per session
4. structure of the curriculum so that every third year student could participate in all simulation scenarios

As soon as we began developing the course, we scheduled a meeting with the directors of four major course clerkships: medicine, surgery, pediatrics, and obstetrics/gynecology. The goal of the meeting was to ensure continuity between the future simulation curriculum and the existing clerkship curricula, as well as to define what medical conditions pertinent to each clerkship should be covered in the simulation program. Also it was decided that each clerkship would be assigned a particular day of the week for its students to attend simulation sessions. This assignment facilitated scheduling and administration of the sessions.

To accommodate the developing third year curriculum along with the existing simulation program for the fourth year medical students rotating with Anesthesiology, we scheduled two sessions per day, each two hours long. In determining the optimal size of a group, we used our experience in working with the fourth year medical students. We believe that the optimal group size is from two to four students. This group size is large enough to provide the students with peer support, yet small enough to keep every participant active during the session. As a final result, approximately 350 simulation sessions covering eight medical conditions were scheduled, such that every medical student could participate in all eight simulation scenarios over the course of his or her third year.

Scenario Content

Determining the design and content of the simulation sessions is a complex and multifactorial process. Creating scenarios for a specific curriculum, one needs to take into account:

1. type of pathology
2. students' level of knowledge [1], [3]
3. learning objectives of the session
4. expected critical performances of the session
5. technical capabilities of the HPS

In developing the curriculum, we decided that some of the scenarios could be interconnected rather than isolated. For example, since eclampsia may be complicated with the development of ARDS, we decided to include both scenarios into the obstetrics/gynecology block.

The medical conditions chosen were spread in the following fashion among four major clerkships:

1. Introductory session with a scenario on thermal injury, upper airway burn, and smoke inhalation.
2. Medicine
   a) Atrial fibrillation
   b) CHF
3. Pediatrics
   a) Severe bronchial asthma attack
   b) Septic shock
4. Surgery
   a) Upper GI bleeding with hemorrhagic shock
   b) Multiple trauma with tension pneumothorax
5. Obstetrics/gynecology
   a) Eclampsia
   b) ARDS

Figure 1 Students find left femur fracture during the "Multiple trauma with tension pneumothorax" scenario

It is important to have an effective introductory session prior to the beginning of the regular simulation sessions. To accomplish this, every third year medical student is sent a web link to the brief web-based introduction to the simulation lab. The web-based introduction describes what the students are expected to do in the sessions and briefly describes the functional capabilities of the HPS. Having no prior exposure to the HPS, some of the students might believe that simulation sessions are the same as small group lectures with the simulator used as a visual aid. We have told the students to approach the simulator exactly the same way they would approach a real patient. Students are instructed to discuss questions relating to patient management with their team members first, then use reference books and personal digital assistants (PDAs), and finally ask the instructor for help as a last resource.

During the introductory session the students receive hands-on experience working with the simulator. They assess the "patient's" vital signs, perform endotracheal intubation, and perform CPR with feedback of the effectiveness of the chest compressions. There is also a brief scenario of thermal injury with upper airway burn and smoke inhalation included in the introductory session. Our experience demonstrates the importance of including a small interactive scenario in the introductory session. The ability of the HPS to respond physiologically and realistically facilitates the transformation of students from passive observers to active caregivers. Effective use of the interactive features of the HPS reinforces the perception that it is the simulator and not the facilitator who responds to the interventions the students perform.

Learning Objectives

The learning objectives, critical performance indicators, level of prior knowledge, and technical capabilities of the simulator are closely interrelated. One of the greatest features of the HPS is its level of interactivity. Unlike the other simulation systems which are predominantly procedure oriented, the HPS provides the greatest amount of interactivity in the current state of simulation development. By the third year of medical school, students have sufficient knowledge of pathophysiology, pharmacological and non-pharmacological methods of treatment, and principles of interpretation of patient response to treatment. This level of knowledge matches perfectly with the technical capabilities of the HPS. In attempting to create a bridge between this knowledge and clinical application, we put great emphasis on the sessions' interactivity. In other words, students are expected to assess the efficacy of their treatment based on the simulated patient's response rather than on the facilitator's comments. The following simplified chart demonstrates the logical relationship between monitoring the patient and therapeutic decision making.

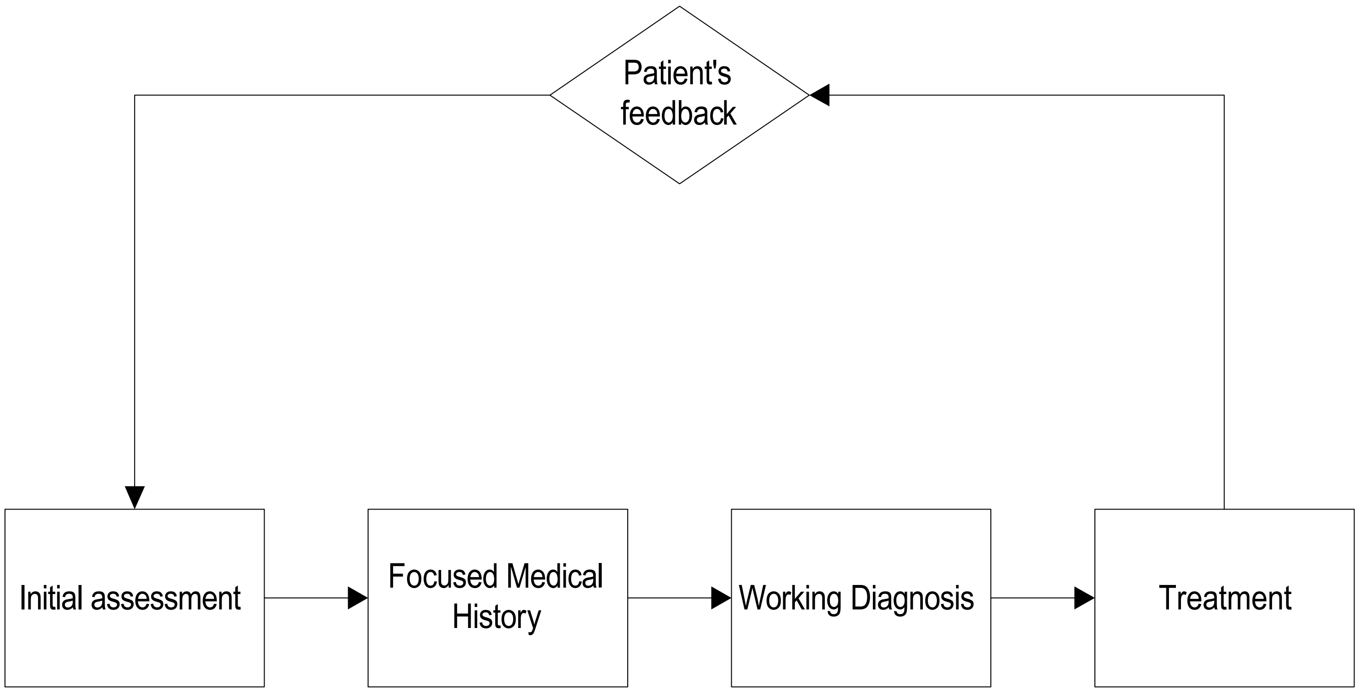


Figure 2 Logical model of scenarios

The following table lists the major learning objectives of each scenario.

Scenario	Learning objectives
Asthma	• Focused history and diagnosis • ABC, oxygen administration • β_2-agonist administration • Steroid administration • Correction of mucus rheology • Fluid administration • Interpretation of pathophysiologic changes • Drug dosage calculation
Septic shock	• Focused history and diagnosis • ABC, oxygen administration • Fluid challenge • Vasopressor administration • Antibiotic administration • Differential diagnosis with hypovolemic shock • Fluid and dopamine resistance recognition and interpretation • Prediction of developing possible complications • Drug dosage calculation
Atrial fibrillation	• Focused history and diagnosis • ABC, oxygen administration • ECG interpretation • Ventricular rate control • Anticoagulation and control of its effectiveness • Rate control • Rhythm maintenance • Recognition of possible complications (pulmonary embolism, ischemic stroke) • Recommendations to the patient
CHF	• Focused history and diagnosis • ABC, oxygen administration • Right-sided vs. left-sided heart failure • Systolic vs. diastolic dysfunction • Preload correction • Afterload correction • Positive inotropic drugs in case of systolic dysfunction • Recommendations to the patient
Upper GI bleeding and hemorrhagic shock	• Focused history and diagnosis • ABC, oxygen administration • Airway protection and prevention of aspiration • Estimated blood volume loss • Fluid replacement rate and volume • Composition of fluid replacement therapy (crystalloids, colloids, blood products) • Choice of vasopressor and its rate of administration • Possible complications and interventions
Multiple trauma and tension pneumothorax	• Focused history and diagnosis • ABC, oxygen administration • Monitoring the vital signs • Tension pneumothorax recognition • Tension pneumothorax decompression • Correction of traumatic and hemorrhagic shocks • Possible complications and interventions
Eclampsia	• Focused history and diagnosis • ABC • Dosage of magnesium sulfate and convulsion control • Clinically assessing the effectiveness of convulsion control • Recognition and correction

	of magnesium toxicity • Hypertension control • Possible complications during and after delivery
ARDS in obstetric practice	• Focused history and diagnosis • ABC, oxygen administration • Differential diagnosis with cardiogenic pulmonary edema • Endotracheal intubation and mechanical ventilation • Ventilator parameter settings • Parenteral feeding

Table 1 Learning objectives of the scenarios that comprise the junior medical school simulation curriculum

Session Administration

The LSU School of Medicine is in a unique position in terms of effective delivery of simulated sessions to its medical students because it owns a state of the art Health Science Center building of approximately 2,500 square feet. This facility houses two full-size operating rooms designated for simulation, two practical skills labs, eight small session rooms fully equipped with high-speed Internet access, and a lecture hall.

Each session begins with a written case presentation that describes the patient's medical background and medication history. After that, the students are reminded that they are physicians and that they are responsible for making decisions to treat the patient. In most of the cases, the facilitator tells them, "I will help you with the treatment of this patient, but whenever you want me to do something, be as precise as possible." For example, in the hemorrhagic shock scenario, an order of "two large bore IVs wide open" will not be accepted. The students must not only order two large bore IVs but must also specify the type of IV fluid and the rate of administration.

To reinforce correct decision making in patient management, students are challenged to verbalize the rationale behind any decision they make. This usually triggers discussion among the team members with recollection of pathophysiologic and pharmacologic concepts important to the case, thereby facilitating peer-to-peer teaching and teamwork. Often, when students approach a critical learning objective and make a correct decision, emphasis is placed on this learning objective by increasing the stress level. For example, when the students decide to administer a glucocorticoid in the bronchial asthma case, the facilitator may assume the role of a child's parent who resents hormone administration unless it is proven to him to be necessary [2] have found this approach to be very effective because it initiates another cause-effect reasoning loop (see Figure 1) and forces the students to verbalize all their decision- making steps. Thus, the students are performing the same scenario several times: once actually performing it, and several more times "replaying" the scenario when they verbalize their decision-making steps.

At the end of simulation session, there is a short debriefing during which the students are asked to narrate the patient's medical history, physical findings, and the major components of that patient's treatment. The session ends with the students taking the same ten multiple choice question quiz. The results of the quizzes are kept in a database for analyses of the effectiveness of the simulation sessions.

Efficacy Measurement

For each simulation session, students take both pre-activity and post-activity quizzes to assess how well the learning objectives are achieved in the session [4]. The same quiz of ten multiple choice questions is given both before and after the session. The quizzes also help the students to focus on the key points of the session.

During the period from the first of July till the first of September, there were one hundred and thirty eight pre-activity and post-activity paired scores obtained for four of eight simulation scenarios. The very first simulation session in each rotation has been devoted to the introduction, and the introduction has not been scored. The results of preliminary statistical analysis and pre/post-test hypothesis testing are encouraging and support learning effectiveness of interactive simulation-based teaching/learning for the targeted clinical emergencies ($N=138$, $p<0.0001$).

Conclusion

Developing a simulation curriculum for medical school students is a complex and multifactorial process. It requires organizational preparation in coordination with existing curricula, identification of scenario content appropriate for students' knowledge level and the simulation system's technical capabilities, identification of learning objectives for each scenario, promotion of interactive learning in the administration of sessions, and the development of tools to measure the effectiveness of these sessions. Additionally, effective administration of each session requires a facilitator with good communicational skills and adequate knowledge of basic science and clinical concepts.

The LSU School of Medicine – New Orleans has developed a Human Patient Simulation curriculum for its junior medical students, and our preliminary statistical analysis shows effectiveness of the simulated sessions as an interactive tool in clinical bed-side teaching.

References

Bradley P. & Posthlestwaite. Medical Eucation. Simulation in Clinical Learning. *Setting up a clinical skills learning facility*. pp. 6-13. Blackwell Publishing Ltd, 2003

McLaughlin, Steven. A., et al, *Human Simulation in Emergency Medicine Training: A model Curriculum*. Academic Emergency Medicine, 2002; 9:1310-1318.

Gruppen,Larry. D., & Frohna, Alice. Z., International Handbook of Research in Medical Education, *Clinical Reasoning,* Dordrecht: Cluwer Academic Publishers 2002. pp. 205-229

Regehr, Glenn, International Handbook of Research in Medical Education, *The Experimental Tradition,* Dordrecht: Cluwer Academic Publishers 2002. pp. 5-43

Olfaction Displays in Virtual Reality

Carl M. Anglesea

Institute for Simulation & Training (IST)
University of Central Florida
Orlando, Florida 32826
canglese@mail.ucf.edu

Abstract

The incorporation of olfactory stimulus in multimedia displays has been the dream of visionaries like Aldous Huxley and avant-garde filmmakers like Woody Allen. Recent development efforts have spawned many olfaction display devices, which can be adapted for use in virtual reality training scenarios. This discussion will begin with a primer on the physiology of olfaction, attempted classification scheme for odorants, and a historical review of the design and use of olfaction display technology. We will conclude with a list of prospective uses for olfaction displays in virtual reality training applications.

Introduction

> *"The scent organ was playing a delightfully refreshing Herbal Capriccio-rippling arpeggios of thyme and lavender, of rosemary, basil, myrtle, tarragon; a series of daring modulations through the spice keys into ambergris; and a slow return through sandalwood, camphor, cedar and newmown hay (with occasional subtle touches of discord – a whiff of kidney pudding, the faintest suspicion of pig's dung) back to the simple aromatics with which the piece began. The final blast of thyme died away; there was a round of applause; the lights went up"* (Huxley, 1932).

Aldous Huxley's scent organ was designed to entertain audiences at the "feelies," a projection of the future state of entertainment in his *Brave New World*. Although his prediction of the future has yet to materialize, recent developments in computer controlled olfaction display devices are bringing virtual reality closer to his vision. We will review the challenges of developing olfaction displays and present areas for near term research.

Physiology

The human smell and taste senses depend on complex, interconnected, chemical detectors that sample and analyze molecules from the environment. In the case of smell, these chemical detectors lie in two small patches of cells called the olfactory epithelium located at the very top of the nasal cavity. Each half of the olfactory epithelium contains on the order of five million olfactory neurons. Each neuron has at least ten hair-like cilia attached to it. These cilia protrude into a thin layer of mucus on the cell surface. It is in this mucus layer that odorant molecules are initially trapped and analyzed by proteins on the cilia. Each olfactory neuron is connected through a hole in the bone above it, called the cribiform plate, to a part of the brain called the olfactory bulb. The size of the olfactory bulb is proportional to the degree of olfactory acuity in the species. The olfactory bulb in turn connects to the smell and taste sensory cortex. The workings of the olfactory bulb still remain a mystery as do the workings of another olfactory sensor called the vomeronasal organs (VNOs). If the VNOs in humans work the same way they do in rats, they would connect to an auxiliary olfactory bulb that connects to a part of the brain controlling maternal behaviour and reproduction, bypassing the cerebral cortex. If this alternative olfactory system exists in humans, we would have no conscious awareness of it (Pines, 1995).

Comparison to Other Senses

Our visual senses respond to electromagnetic energy that can be measured and reproduced on a convenient linear scale. So too can the pressure waves of sound be measured and reproduced in linear fashion. However, the first challenge faced in the design of olfaction displays is the lack of a convenient linear scale on which to place the odor producing chemical substances which we will refer to as odorants. The number of individual odorants detectable by humans is estimated at about 10,000. As with the other senses olfaction acuity can be measured and classified: macrosmia (a good sense of smell), microsmia (a poor sense of smell), anosmia (a loss of the sense of smell), and parosmia (a perverted sense of smell) (Eisenberg, 2002).

Classification of Odorants

Since the time of Linnaeus, countless attempts have been made to classify odorants. All of the classifications reflect more about the cultural and social bias of the classifiers than the qualities of the odorants themselves. The two latest classifications are based on biochemistry and reflect more about the possible mechanisms of olfaction. The stereochemical theory classifies odorants by their molecular shape, dividing the population into seven categories (Amoore, 1964):

- Ethereal (dry-cleaning fluid)
- Camphoraceous (camphor)
- Musky (angelica root oil)
- Floral (roses)
- Pepperminty (peppermint candy)
- Pungent (vinegar)
- Putrid (rotten egg)

Most scents can be explained using Amoore's theory, but many cannot. A more recent classification is based on vibrational theory which reflects the molecular structure of the odorants and explains the exceptions to Amoore's theory (Turin, 1996).

Display Design and Use

A review of prior attempts to use olfaction displays in the entertainment industry reveals possible implementation pitfalls for VR. The first serious attempts were made in the late 1950s when cinema owners experimented with 3-D glasses, vibrating seats, and olfaction devices to lure theatregoers away from the rising popularity of television (Kaye, 2001).

AromaRama and Smell-O-Vision

The first such device, AromaRama, used freon gas to disburse odorants through the cinema's air conditioning system, attempting to immerse the audience in a travelogue through China. The odorants were not realistic, the system provided little control over their intensity or removal, and the film was critically panned. The second device, Smell-O-Vision, was a more advanced system, which delivered odorants to the individual seats. This system also suffered from the use of unrealistic odorants, which were easily detected and critically rejected after only one attempted film, a murder mystery. The story writers must be credited with some creativity, though, for using an olfactory clue, the smell of pipe smoke at the crime scene, to engage the audience in the storyline.

Sensorama

The best early attempt at a multi-sensory virtual reality simulation was designed in the early 1960s by Morton Heilig (Heilig, 1962 in Kaye, 2001). About the size of a large video game, Sensorama was a motorcycle riding simulator that incorporated vibration sensing in the seat, fans to produce wind, and odorants released when passing pizza parlours and roadside flower gardens. The system was never commercially funded, however, despite years of development.

Figure 1: A Sensorama kiosk

Scratch-n-Sniff

The next few decades saw few serious advances in olfaction displays with the notable exception of scratch 'n sniff technology developed by the 3M Corporation. 3M encapsulated fragrances in a polymer coating that can be printed on paper and released by scratching the coating. Scratch-n-sniff cards played a frivolous role in the 1981 film, *Polyester*, but are used mostly for the advertisement of perfume samples. They do demonstrate the ability to store small amounts of odorants in a safe, compact package, which can be easily released when needed. An interesting use of Scratch-n-sniff came from Switzerland which issued a chocolate scented postage stamp in September 2001.

Figure 2: Swiss Postage stamp

D.I.V.E.

The record of recent attempts begins with John Cater at the Deep Immersion Virtual Environment Laboratory of the Southwest Research Institute who designed an odorant delivery system for fire fighter training. Cater describes the early version of the system as a backpack mounted automatic scratch-n-sniff player that can deliver variable intensity odorants to the participants mask with a ¼ second response time. "Odors range from burning wood, grease and rubber to sulphur, oil and diesel exhaust. Lifetime of the odor cartridges is 6 months to a year without refilling. Olfactory output is…completely proportional from a hint of odor to a stench that makes you want to rip the mask off…adds about $5000 to the cost of a VR system (Cater, 1992 in Kaye, 2001)."

Recently Patented Devices

A list of recent advances shows the level of interest and state of olfaction display development and integration:

- De Sousa describes a personal use olfactory dispenser similar in form to a compact disk player which uses a CD like disk containing embedded fragrance capsules (de Sousa, 1999 in Kaye, 2001).
- Prendergrass adds computer control to his device and claims it is "'especially useful for providing a realistic sensory experience in an interactive or non-interactive use, and may be used in…the entertainment industry, the educational training field or a medical arena' and includes a virtual reality helmet and a neck-mounted individual smell output device (Prendergrass, 1996 in Kaye, 2001)."
- Martin and Barbier further refine their devices incorporating breath sensors to time the odorant release and pressurized oxygen to speed delivery. At this point we should mention the chemical safety considerations for these devices. As Barbier states, "an over oxygenation of a zone or of a medium may indeed cause an activation of combustion phenomena, resulting in a degradation of some materials such as electric motors," not to mention the wearer (Martin 1997, and Barbier, 1998 in Kaye, 2001).
- Corporate America enters the field led by IBM's "Computer Controlled Olfactory Mixer and Dispenser for Use in Multimedia Applications" and "Aroma Sensory Stimulation in Multimedia." Motorola holds the rights to a smell output device in a PCMCIA or PC adapter card and an Israeli company, Aromix, holds "Methods and Apparatus for Odor Transmission" and "Methods and Apparatus for Odor Reproduction" (Kaye, 2001). Can a wireless odornet be far away?
- Trisenx becomes a dominate player in the current olfaction device market, selling a software controlled 60 aroma Scent Dome for $269 and a single scent sampler device for $1.32 in bulk from their web site: www.trisenx.com.
- IST issues the Virtual Environment Software SandBox (VESS) 3.0.0 User's Guide supporting olfaction devices and describing the hardware interface and software API for placing odors in a virtual environment (vess.ist.ucf.edu).

Research Applications

The previous discussion of olfaction devices points to the availability of cost effective olfaction displays for use in virtual reality training applications. The following discussion suggests uses for olfaction displays in virtual reality training applications.

Memory and Recall

The most promising application of olfaction displays may be the enhancement of memory and recall of skills learned in a simulation environment. The now classic description by Marcel Proust of long forgotten childhood memories evoked by the aroma and taste of tea soaked madeleine highlights the relationship between olfaction and recall (Proust, 1913). Improvements in recall may be accomplished by producing an ambient smell in the simulator training environment that will be present in the real world environment, thereby triggering recall. Several studies have found that improvement in memory resulted when the same ambient odor was present at both learning and recall (Schad 1990, Cann and Ross (1989), and Smith etal 1992 in Kaye 2001). A special case of memory recall training exists in the nursing and medical fields. Odors given off by the patient's body can provide diagnostic clues to medical professionals.

Presence and Immersion

The attempts to introduce olfaction in the entertainment field, described previously, were intended to immerse the audience in the film experience. One reason they may have failed is due to the diminishing returns associated with combining input from multiple senses. When additional sensory input is added to existing sensory input without providing additional information, the more dominant sense will take over and the weaker sense can be ignored. The weaker sense can then be criticized for any discrepancies with the information provided by the more dominant sense. Olfaction displays may prove useful in increasing the sense of presence by simply masking any ambient odors in the simulator environment.

Mood and Emotion

As discussed in the physiology section, olfaction is a powerful, subliminal sense that operates in part at the subconscious level. Intentional or not, the addition of olfactory displays to a simulation has the potential to bring emotion, mood changes, and subconscious reactions to the participant. A careful study should be made in the selection of odorants and their predicted effects in a simulation.

Abstract Information Display – Smell Icons

As mentioned in the discussion of odorant classification, associating smells with information is learned and reinforced with each whiff. So, associating or reassociating a smell with an abstract piece of information is easy to do. This is not a new idea. Incense clocks have been part of Chinese and Japanese temple rituals for a thousand years (Bendini 1964). The clocks, designed to burn a different incense each hour, allowed temple priests to tell the time to within one hour (the message: jasmine smell = it's after noon). Smells are also used for alarm applications when no better indication is available. Since its inception, the natural gas industry in the United States has included small amounts of hydrogen sulfide gas in the odorless natural gas pumped into home communities to trigger alarm reactions in people who encounter leaking gas (the message: sulphur smell = move away!). North Sea mine networks are prepared to pump the scent of wintergreen into their extensive ventilation systems in the event of a catastrophe (the message: wintergreen smell = evacuate the mines!). Joseph Kaye at the MIT Media Lab has experimented with a two-aroma indicator device to notify lab visitors about the current status of the stock market. Dubbed Dollars & Scents, the computer controlled device emits the smell of mint when a selected market indicator is rising and the smell of lemons when it is declining (the message: mint smell = we're making money) (Kaye, 2001). Kaye's thesis work also proposes the use of smells as icons indicating the state or condition of a system in the same way that a computer's icons indicate its battery charge level or network connection status.

Conclusion

If the recent flurry of interest in olfaction display design is any indication, these devices will soon become more readily available in the general population as well as virtual reality. Huxley's scent organ will likely remain a fantasy, but the new devices will undoubtedly provide interesting research in the near future.

References

Amoore, J. E., Johnston, J.W., Rubin, M. (1964). The Stereochemical Theory of Odor. Scientific American. **210**.

Bendini, S. A. (1994). "Holy Smoke: The Oriental Fireclocks." New Scientist **21**: 537-539.

Eisenberg, A. M. (2002). Olfaction: A Primal Legacy--Part I. Croton on Hudson, NY, Dynamic Chiropractic.

Huxley, A. (1932). Brave New World.

Kaye, J. N. (2001). Symbolic Olfactory Display. Media Arts and Sciences. Cambridge, MA, Massachusetts Institute of Technology**:** 1-124.

Pines, M. (1995). Seeing, Hearing, and Smelling the World. Chevy Chase, MD, Howard Hughes Medical Institute**:** 46-56.

Proust, M. (1913). Remembrance of Things Past. New York, Vintage.

Turin, L. (1996). "A Spectroscopic Mechanism for Primary Olfactory Reception." Chemical Senses **21**(773).

Simulation and Training

Efficacy of Transfer in Simulation-Based Training: Implications for Stress Exposure Training

Jennifer M. Ross, James L. Szalma, & Peter A. Hancock

Psychology Department
And Institute of Simulation and Training
University of Central Florida
Orlando, FL

Abstract

Transfer of training from simulators to the real-world has recently come under investigation as the generalizability of task-specific training has come into question. New hypotheses recommend that, to ensure effective performance under stress in real-world environments, one should supplement skill-based training with stress exposure training. Stress exposure training has further benefits in that it may serve as a more generalizable form of training and transfer across tasks and stressors. The impact of improving performance and reducing perceived stress and workload is of vital importance to many military operations, especially in high technology and high workload situations such as the Landwarrier or Unmanned Arial Vehicles (UAVs), in which mistakes are costly in terms of economics as well as life. In this article the premise of and future implications of Stress Exposure Training are discussed.

Simulation Training

Simulation has been successfully employed in military and commercial sectors to train individuals (e.g., pilots, soldiers, doctors) with the objective of increasing their performance accuracy in real-world situations. The primary method of performance improvement in these contexts is to provide simulator-based practice of situations and conditions similar to those expected to be encountered in a real-world Stress Exposure Training. However, the extent to which simulator-based training transfers to real-world application has recently been questioned (Driskell, Johnston, & Salas, 2001). New theories of simulator training challenge traditional task-based simulation training, claiming that skill acquisition is highly task-specific and not readily transferable to different tasks and situations (Sims & Mayer, 2002). This limitation decreases the applicability of simulator training transferring to performance in a real-world Stress Exposure Training, which is highly complex, dynamic, and virtually impossible to anticipate all potential task/accident scenarios that may occur. In a review by Ivancevich and colleagues (1990), recommendations were made that training programs should be designed to address performance outcomes associated with specific stressors. Indeed, such disasters as Three Mile Island, Chernobyl, and the USS Vincennes have underscored the importance of developing training interventions to offset the impact of stressors on complex cognitive tasks (Johnston & Cannon-Bowers, 1996).

Stress Exposure Training

One avenue of increasing the generalizability of simulator-based training has been through empirical work which has been conducted on Stress Exposure Training (Klepac, Hauge, Dowling, & McDonald, 1981; Saunders, 1993; Saunders, Driskell, Johnston, Salas, 1996). Considerable evidence has shown that stressors cause decrements in attitudes and performance, which are related to outcomes that are emotional, behavioral, or physiological in nature (Greenhaus & Parasuraman, 1987). Stress Exposure Training works by training individuals on factors that are more generalizable than task skills, such as controlling perceived stress and workload by providing intervention information on stressors, sensory information on physiological and emotional reactions to stress, and procedural information on how stress effects task performance. These three phases of stress exposure training will be further discussed later in the paper.

History Stress Exposure Training

Since the inception of stress exposure from Wolpe's work (as cited in Johnston & Cannon-Bowers, 1996) and eventually cognitive/behavioral stress-coping training in the early 1970s, the application of stress exposure training has generally been clinical, and applied chiefly to uses, as alleviating physical pain, anxiety, depression, and anger (Deffenbacher & Suinn, 1988). Gradually, stress-coping training was expanded to include training to a number of occupations (e.g., nurses, police officers, oil-rig trainees), and to enhance athletic performance (Johnston & Cannon-Bowers, 1996). In a meta-analysis by Saunders, Driskell, Johnston, and Salas (1996; See also Saunders, 1993) it was demonstrated that this type of training is an effective means for reducing state anxiety, reducing skill-specific anxiety, and enhancing performance under stress.

Johnston & Cannon-Bowers (1996) argues that there is not sufficient research to suggest that training developers adopt the strategies used in cognitive/behavioral stress coping training programs. This was due to two reasons: first traditional stress exposure research had too narrow a focus (clinical and sports), and second, little work had been done to integrate research findings into a conceptual framework to guide the development of incorporating stressors into training and examining coping skills training for complex cognitive tasks. Therefore, they advocated the use of the term Stress Exposure Training to extend the focus of 'cognitive/behavioral stress coping' beyond its original clinical domain.

Stress Exposure Training has three main objectives, 1) to build skills that promote effective performance under stress, 2) to build performance confidence, and 3) to enhance familiarity with the stress environment. According to Meichenbaum (1985) these three objectives can be accomplished by employing two components: 1) stress-coping training – features developing skills that reduce potential cognitive and psychomotor performance deficiencies resulting from specific stressors (Driskell & Salas, 1991) and 2) instructional design – features a process of gradual exposure to realistic stressors that enhance learning of coping skills, and is based on basic principles of training design for skills acquisition. The two integrated components have resulted in a three-phase process (See Table 1; Meichenbaum, 1985).

The Three Phases of Stress Exposure Training

Phase 1. The first phase in stress exposure training typically involves a discussion of common reactions people have to specific stressors they encounter, with the objective to help the individual understand that they will be able to stop the negative thoughts and behaviours that contribute to stress. It is expected that as the individual's anxiety is reduced, perceived control increases, and consequently ones efficacy in dealing with problem stressors is improved (Johnston & Cannon-Bowers, 1996). Interestingly, violations of expectations have been found to be universally detrimental to performance, even when participants' expectations are violated by their situation being better than expected (Keinan & Friedland, 1996). The effect of credible prepatory information on performance has been found to result in reduced subjective stress, enhanced task confidence, and fewer errors (Inzana, Driskell, Salas, & Johnston, 1996).

Phase 2. The second phase focuses on learning stress coping skills through practice and feedback. The objective is to train the individual to maintain an awareness of stress reactions in order to invoke appropriate skills to reduce stress. The act of maintaining an awareness of thoughts and actions is referred to as "metacognition" (Glaser & Bassok, 1989). Therefore, metacognitive skills; such as, thought restructuring (replacing negative thoughts and reactions that are triggered by a stressor with positive coping thoughts and reactions), problem solving (reduce task performance errors), & physiological control (involves deep breathing and muscle relaxation methods to calm physiological reactions to stressful encounters) support the execution of competent performance.

Phase 3. The third phase involves practicing the coping skills in a setting that simulates or reproduces the problem stressors. Ways of coping with stress are often classified into three areas: task-focused coping, emotion-focused coping, and avoidant-focused coping (Lazarus & Folkman, 1984). Task-focused coping circumvents negative affect through cognitive and behavioural activity that minimizes or modifies the stressor. In this case the individual focuses on dealing with the task and develops strategies aimed at accomplishing the goals of the task (Thropp, Szalma, Ross, & Hancock, 2003). Emotion-focused coping consists of attempts to alleviate or eliminate emotions elicited by a stressor directly, with little attention paid to the characteristics of the situation or to the nature of the threat it poses (Keinan & Friedland, 1996). While, emotion-focused coping has been found to be effective in preparing individuals to cope with stressors that they cannot control and which are brief in duration, it has proved less effective/relevant for training people to withstand and cope with combat stressors. This more prolonged and controllable situations requires mixed or problem-focused coping. The third coping strategy, avoidant coping, involves diverting one's attention from the stressful situation (see also Cox & Ferguson, 1991). Expected outcomes from this phase are reduced anxiety, increased perceived efficacy toward performance, and improved cognitive and psychomotor performance. Further research involves examining phased-training procedures that divide task acquisition without stress exposure from practice of newly acquired skills under stress. Other methods of skill practice under stress include graduated-intensity training (Keinan & Friedland, 1996).

Table 1: Stress Exposure Training Design (as cited from Johnston & Cannon-Bowers, 1996).

	Stress Exposure Training Design Phase 1	Phase 2	Phase 3
	Presentation of requisite knowledge	Skill practice with feedback	Skill practice with stressors
Objectives	• Knowledge of typical reactions to stressors.	Develop metacognitive skills • Positive coping thoughts and behaviors • Use relaxation techniques to calm physiological reactions • Develop cognitive skills • Use problem solving skills	• Use phase 2 skills while exposed to stressors.
Outcomes	• Increased perceived efficacy in dealing with stressors.	• Reduced Negative attitudes toward self and stressors. • Increased use of positive thoughts and behaviors. • Reduced blood pressure, heart rate, and increased psychomotor steadiness. • Successful coping skill performance.	• Reduced anxiety. • Increased efficacy. • Successful application of skills while exposed to stressors. • Improved cognitive and psychomotor performance under stress.

Theory behind Stress Exposure Training

One can examine the mechanism of Stress Exposure Training through the transactional model of stress. In this view stress occurs when the perceived demands of the situation tax or exceed the perceived resources of the individual to meet those demands (Lazarus & Folkman, 1984). Based on this theory, Meichenbaum (1993; as cited in Johnston & Cannon-Bowers, 1996) has proposed that Stress Exposure Training provides the skills that should reduce the imbalance between the demands of the stressful situations and the individual's coping resources. The benefits of decreasing this discrepancy have a far-reaching impact on stress and performance.

One can further, examine the relationship between stress and performance by referring to the Hancock and Warm dynamic model of stress and sustained attention (1989). In this model, performance is directly influenced by workload and stress factors, where stress is integrally related to environment (Hockey, 1983), appraisal/coping mechanisms of the exposed individual (Lazarus, 1966), and the general response of the physiological system (Selye, 1956; see Figure 1). As these factors combine and lead to debilitative levels of hyper- or hypo-stress the operative state begins to degrade first psychologically, in increased perceived stress and workload, and then physically through impaired performance. Support for this view has been found across two meta-analyses. In a meta-analysis, by Driskell and colleagues, the effects of stressors on self-reported stress and performance was examined, and it was found that people were more likely to report "feeling stressed" before showing evidence of performance impairment (as cited in Saunders, 1993). In a separate meta-analysis conducted by Saunders (1993) moderate magnitude effects were found for stress inoculation training (a similar three phase stress exposure training methodology) for state and skill based anxiety, but only small magnitude effects for performance. From this Saunders drew the conclusion that subjective reports of stress were more sensitive to the effects of stress inoculation training than performance; that is, reports of lower perceived stress would occur before signs of performance improvement.

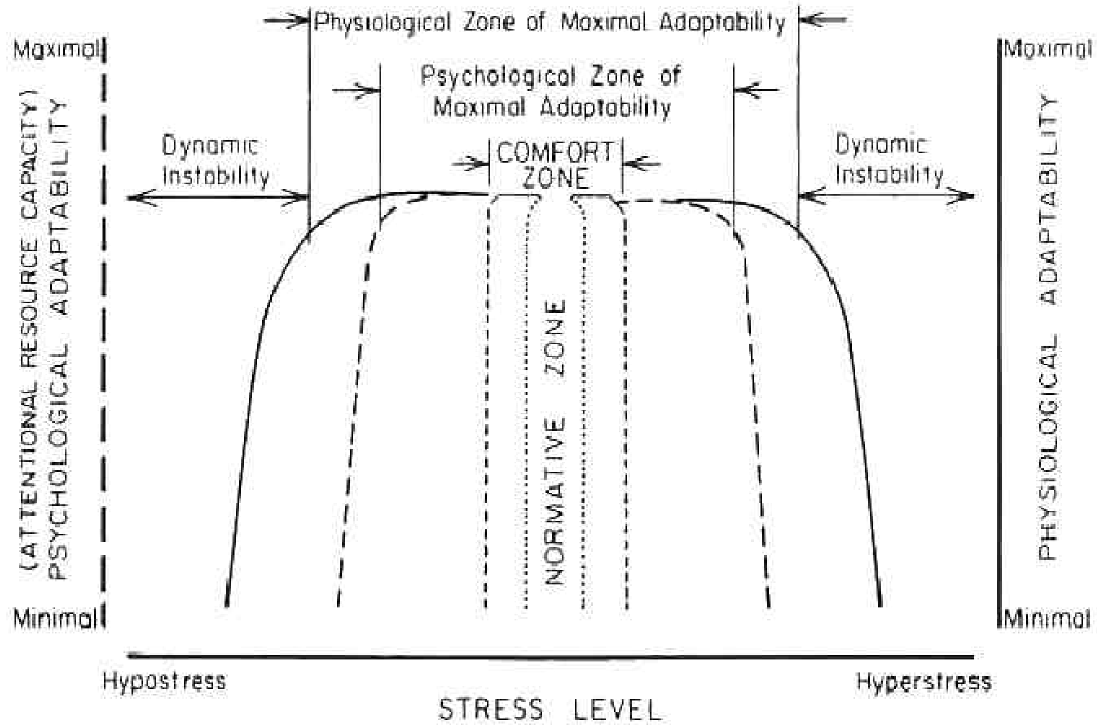


Figure 1: Hancock and Warm dynamic model of stress and sustained attention (as cited from Hancock & Warm, 1989).

New Directions

Research examined in the meta-analysis conducted by Saunders and colleagues (1993, 1996) has supported the view that stress impacts observers psychological before performance begins to degrade. While, these hypotheses support the Hancock and Warm model of dynamic stress and sustained attention, it would be critical to examine the degree to which task-based environmental factors and coping strategies as moderators of Stress Exposure Training efficacy. It is further imperative to investigate limits of transfer of training between field and laboratory training. Studies are currently under investigation to examine the effects of spatial-temporal perceptions on performance and perceived stress/workload with and without stress exposure training. The effects of individual differences in coping strategy will be used in regression to examine the moderating factors of this variable on Stress Exposure Training efficacy. Finally, while Saunders (1993) found Stress Exposure Training to beneficially affect performance in the lab and the field, they did not examine the transfer of Stress Exposure Training from a lab to a field task. Indeed, as simulations used in training environments become more realistic the question of specificity becomes more important. That is as realism is improved; this could lead to lower generalizability and hampered transfer of training. These factors represent new avenues of research that must be examined to fully develop the construct of Stress Exposure Training. Figure 2 represents a proposed model to test Stress Exposure Training as related by the moderating variables of environmental factors (i.e., spatial or temporal task demands), individual differences in coping style, and training transfer.

Conclusion

The generalizations of training effects from one task/stressor to another are of critical concern. Future research must examine how Stress Exposure Training principles transfer from the training to the operational environment, and which factors facilitate or inhibit this transfer. Clearly, the cost of training might be prohibitive if research shows that the effectiveness of stress training is limited to specific stressor-task combinations.

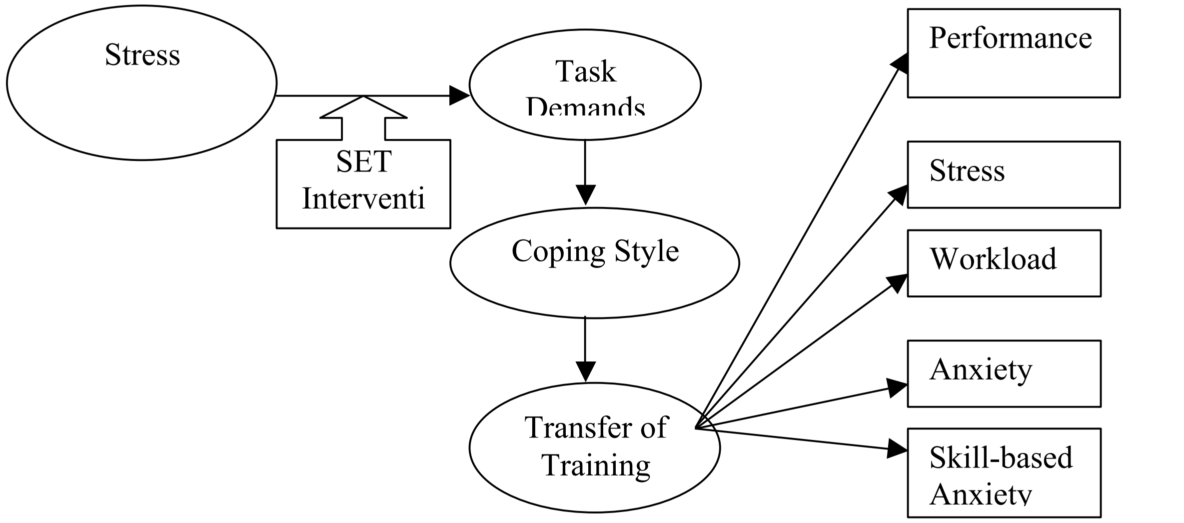


Figure 2: Hypothesized relationship between Stress Exposure Training and Moderators of environmental task demands, individual differences in coping style, and training transfer.

References

Cox, T & Ferguson, E (1991) Individual difference, stress and coping. In Cooper , C and Payne, L (Eds) *Personality and stress: individual differences in the stress process.* (pp 7-29). Chichester, UK: Wiley & Sons.

Deffenbacher, J.L., & Suinn, R.M. (1988). Systematic desensitization and the reduction of anxiety. *The Counseling Psychologist, 16(1),* 9-30.

Driskell, J.E., & Johnston, J.H. (1998). Stress Exposure Training. In J.A. Connon-Bowers & E. Salas (Eds.), *Making decisions under stress: Implications for individual and team training* (pp. 191-217). Mahwah, N.J. : Lawrence Erlbaum Associates.

Driskell, J.E., Johnston, J.H., & Salas, E. (2001). Does stress training generalize to novel Stress Exposure Trainingtings? *Human Factors, 43(1),* 99-110. Mahwah, N.J. : Lawrence Erlbaum Associates.

Driskell, J.E., & Salas, E. (1991). Overcoming the Effects of Stress on Military Performance: Human Factors, Training, and Selection Strategies. In G. Reuven & A.D. Mangelsdorff (Eds.), *Handbook of military psychology* (pp. 183-193). NewYork, NY: John Wiley & Sons.

Glaser, R., & Bassok, M. (1989). Learning theory and the study of instruction. *Annual Review Psychology, 40,* 631-666.

Greenhaus, J.H., & Parasuraman, R (1987). A work/nonwork interactive perspective of stress and its consequences. Special issue: Job stress: From theory to suggestion. *Journal of Organizational Behavior Management, 8*, 37-59.

Hancock, P.A., and Warm, J.S. (1984). A dynamic model of stress and sustained attention. *Human Factors, 31(5),* 519-537.

Hockey, R. (1983). *Stress and fatigue in human performance.* Chichester, UK: Wiley.

Inzana, C.M., Driskell, J.E., Salas, E., & Johnston, J.H. (1996). Effects of preparatory information on enhancing performance under stress. *Journal of Applied Psychology, 81(4),* 429-435.

Ivancevich, J.M., & Matteson, M.T., Freedman, S.M., & Philips, J.S. (1990). Worksite stress management interventions. *American Psychologist, 45,* 252-261.

Johnston, J.H., & Cannon-Bowers, J.A. (1996). Training for Stress Exposure. In E. Salas (Eds.), *Stress and human performance* (pp. 223-256). Hillsdale, NJ: Erlbaum.

Keinan, G., & Friedland, N. (1996). Training Effective Performance Under Stress: Queries, Dilemmas, and Possible Solutions. In E. Salas (Eds.), *Stress and human performance* (pp. 257-277). Hillsdale, NJ: Erlbaum.

Klepac, R.K., Hauge, G., Dowling, J., & McDonald, M. (1981). Direct and generalized effects of three components of stress inoculation for increased pain tolerance. *Behavior Therapy, 12(3),* 417-424.

Lazarus, R.S. (1966). *Psychological stress and the coping process.* New York: McGraw-Hill.

Lazarus, R.S., & Folkman, S. (1984). *Stress appraisal and coping*. New York: Springer.

Meichenbaum, D. (1985). *Stress Inoculation Training.* New York, NY: Pergamon Press.

Saunders, T.L. (1993). *The effect of stress inoculation training on anxiety and performance.* Unpublished master's thesis, University of Central Florida, Orlando, Florida, United States.

Saunders, T., Driskell, J.E., Johnston, J.H., & Salas, E. (1996). The effect of stress inoculation training on anxiety and performance. *Journal of Occupational Health Psychology, 1(2),* 170-186.

Selye, H. (1956). *The stress of life.* New York: McGraw-Hill.

Sims, V.K., & Mayer, R.E. (2002). Domain specificity of spatial expertise: The case of video game players. *Applied cognitive psychology, 16(1),* 97-115.

Thropp, J.E., Szalma, J.L., Ross, J.M., & Hancock, P.A. (2003). Individual differences in dispositional pessimism, stress, and coping as a function of task type. *Proceedings of the Human Factors and Ergonomics Society, USA, 47,* 1073-1077.

Instructional strategy and aptitude in a driving simulator

Martijn L. van Emmerik

Department of Training and Instruction
TNO Human Factors
PO Box 23, 3769 ZG Soesterberg, The Netherlands
vanemmerik@tm.tno.nl

Abstract

In the search for innovative instruction and feedback concepts for simulator training, this paper provides a description of the final experiment of a series of three that was done to compare the effectiveness of verbal versus non-verbal instructions in a driving simulator.

In the experiment, provision of non-verbal instructions was inspired on the principle of cue-augmentation in which naturally occurring cues are emphasized to draw trainees' attention to the proper cues. It has been suggested that this way of information presentation is more intuitive than verbal presentation and therefore might increase training efficiency.

Using the TNO low-cost driving simulator (LOCS), subjects without driving experience received pre-programmed instructions from a professional driving instructor while they learned to drive a simulated car.

The baseline instructions were verbal recordings of 12 basic comments related to speed of driving, position on the road, steering behavior in curves, gear shifting, and negotiating an intersection. In the non-verbal condition the content of each of these instructions was provided as a visual, aural, or haptic cue.

On the basis of their initial performance on a short introductory scenario in the simulator during which no instruction or feedback was provided, each subject was assigned to either the high or low aptitude group. Aptitude was expected to mediate the efficiency of instruction. Both objective simulator data and subjective data (questionnaires) were gathered.

The results showed a large and significant difference in performance between the two aptitude groups. The differences between the instruction conditions were not significant however. Nevertheless, the subjective data showed that both instructor and trainee generally liked the non-verbal instructions. In the low-aptitude group they were clearly used more frequently than the verbal instructions. This may indicate that non-verbal instructions are easier to provide when necessary. If not, they are used just as much as verbal instructions. Based on the comments of the trainees it is hypothesized that non-verbal instruction is less intrusive than verbal instruction. The trainee more often experiences the latter as 'negative' whereas non-verbal instructions are seen as a hint leaving open the way to self-recovery. Therefore, they may be seen as especially helpful to low-aptitude trainees.

Introduction

This paper describes the results of the final experiment in a series of three that was set up to find support for the claim that simulators offer didactical advantages compared to the operational environment. Based on ideas from the literature and experimental findings (e.g. Lintern, Thomley-Yates, Nelson & Roscoe 1987; Roscoe, 1991; Schneider, 1985) it was hypothesized that deviations from reality could enhance training efficiency on a simulator as compared to training on the operational system. A concept that has been used in this connection is augmented cueing (O'Shea, Cook & Young, 1999; Young, Stedmon & Cook, 1999).

Augmented cueing works by emphasizing elements in the (virtual) environment to help trainees focus on the relevant characteristics of a task. This is particularly relevant during the initial phases of training. This way trainees are expected to understand the task faster which in turn helps speeding up the process of learning. Experimental research on augmented cueing has yielded mixed results, likely as a result from the multitude of aspects affecting instructional efficiency (Lintern & Koonce 1992; O'Shea, Cook & Young, 1999). Clearly, this observation illustrates the relevancy to be aware of possible confounding variables.

Experimental research in the field of training simulation is in fact prone to several sources of confounding. In this light some of the design decisions are explained below.

Although the aim of the experiment was to study the didactical advantages of a simulator compared to the real car it would have been a mistake to test the verbal instruction in a real car and the augmented instructions in a simulator. The differences between the car and the simulator would have introduced a source of confounding making the results inexplicable. As Korteling & Sluimer (1999) point out, it is plausible that a part of the results would have to be attributed to differences in the hardware (car vs. simulator). Which part that would be cannot be determined. This problem was avoided in the present research by using the simulator in both conditions.

Another risk of confounding arises from the differences between instructional content in the conditions. We were interested in the differences between two forms of instruction. If, additionally, differences in content were introduced, this would obscure the experimental results. A fair comparison, therefore, can only occur if the two conditions differ in form only. To assure this, content was kept the same as much as possible. In both conditions instructions and feedback were limited to the same set of actions / mistakes. These were pre-designed, recorded, and assigned to a computer keyboard.

A third possible source of confounding was constituted by 'a priori skill differences'. Because the data of a previous experiment did not lead to unambiguous conclusions with regard to the differences between the two types of instruction, this follow-up experiment was set up with specific measures to control this variable.

To rule out the effect of a confounding variable, however, subjects should be matched on that variable. A first requisite for this is the availability of a preliminary test that can give a reliable indication of the subjects' scores on the matching variable. In the present experiment, the relevant question is 'how can we match subjects on their aptitude for driving the simulator?' Many standard cognitive and psychomotor tests have been proposed and used to give an indication of driving performance (Ball, Owsley, Sloane, Roenker & Bruni 1993; Heikkilä, 2000). A few examples are 'visual acuity, contrast sensitivity, eye health, visual memory, personality questionnaires, (choice) reaction time, and information processing tests. Correlation with driving performance (investigating crashing behavior, Ball et al., or looking at faults and offences, Heikkilä) is generally low although studies with specific groups of drivers have been able to yield higher correlations with driving performance. Still it is difficult to generalize from findings referring to specific groups (people with neurological deficiencies or patients suffering from Parkinson's disease) in order to predict the performance of healthy people who never have driven before. The usability of these measures therefore is questionable. Furthermore, it is known that there are large differences between inexperienced and experienced drivers with regard to their ability to select the appropriate cues and to interpret them correctly (Summala, Lamble & Laakso, 1998).

Method

In this particular experiment subjects were trained to drive a simulated car. They received either verbal instructions or instructions based on cue-augmentation. Both types were activated by keystrokes: in the verbal instruction condition recorded voice-commands were played, and in the augmented instruction condition these commands were translated into a non verbal-instruction (description in Table 1).

An additional aim of this study was to find out if (and how) aptitude differences mediated the effect of the instructions. Therefore, subjects were assigned to different aptitude groups (high- or low aptitude) by means of their performance on a preliminary simulator-driving test. The four groups resulting from this manipulation will be referred to as AI-low or AI-high and VI-low or VI-high (where AI stands for augmented instruction and VI for verbal instruction).

Table 1: Instructions provided in the experiment

Key	Type of Instruction	
	Verbal (recorded voice message)[1]	Augmented
1	'Come on, accelerate'	Roaring engine sound
2	'Not faster than 30'	Traffic sign displayed on screen
3	'Not faster than 50'	Traffic sign displayed on screen
4	'Stop'	Traffic sign displayed on screen
5	'Little bit left'	Position line displayed on the road
6	'Little bit right'	Position line displayed on the road
7	'Attention! Give right of way'	Traffic sign displayed on screen
8	'Look carefully at the intersection'	Three beeps (left-right-left)
9	'Start turning left now'	Jerk at steering wheel
10	'Start turning right now'	Jerk at steering wheel
11	'Gear up'	Constant high 'revs' sound
12	'Gear down'	Constant low 'revs' sound

[1] Translated from Dutch. All messages were kept as short - and to the point- as possible (6 syllables at most in Dutch).

Instrumentation

The present experiment was conducted in the TNO Low Cost Driving Simulator (LOCS). The LOCS is a PC-based simulator for research into simulator training, validity, transfer, and training effectiveness. It also serves as a demonstrator (see Figure 1 & Figure 2).

The LOCS mock-up provides the interface with the simulation. It consists of a car seat, steering wheel, pedals (brake, acceleration, clutch), gear lever, and an (analogue) speed indicator. All components are original car parts (albeit from different cars) except for the speedometer, which is self made.

Figure 1: The TNO-HF Low-cost simulator (LOCS): five displays are placed in a semi-circle, mirrors are represented by the LCD-screens. The LOCS is equipped with a car seat, steering wheel, pedals for acceleration, brake, and clutch, and a 4-position gearbox.

Figure 2: A closer look at the LOCS. Central monitor above LCD-screen with dashboard information (providing speedometer, odometer, rev meter, and indicator lights). Part of the rearview mirror (LCD-screen) is visible.

Furthermore, the LOCS consists of a number of computers that all serve specific purposes. Three Windows NT machines generate the visual environment on five wide screen (24") displays that are positioned in a semi-circular configuration. Each of these computer monitors has a horizontal field of view (FOV) of approximately 40° resulting in a total FOV of about 200°. The central screen displays a high-quality image, which is generated by a single computer. The other two computers each generate a reduced resolution image on two peripheral displays. For that reason, the video-card of the central image generator is more powerful than that of the two other machines.

Finally, one MS-DOS computer ('model computer') receives the signals generated in the mock-up (by standard A/D, DAC, and RS232 interfaces) and uses them to compute the corresponding vehicle behavior mathematical model. This machine provides input to the 'sound computer' (Windows '98), which generates the appropriate engine sounds, and communicates with the 'supervisor computer'. The supervisor is used for scenario control, generation of other traffic, and data storage.

The communication between these different computers is established through an Ethernet connection.

Subjects

Twenty-eight subjects participated in this experiment (11 male, 17 female subjects). Seven subjects were assigned to each experimental group. Because the subjects had to be assigned to the groups based on a preliminary test, it was difficult to match subjects on sex as well as on their test results. In the first place, care was taken that the number of high and low skill subjects was equal for both instruction conditions. With regard to the ratio of male / female subjects in each group it can be said that this was correctly represented in both low aptitude groups. Female subjects, however, were slightly over-represented in the high aptitude - augmented instruction group.

The average age of the subjects was 21.1 years with a standard deviation of 2.7 years (youngest 18 oldest 30). None of the subjects had received driving instruction in a car prior to the experiment.

Procedure

In order to match the subjects on their aptitude to drive a simulator it was decided to have the subjects drive two short routes prior to the experiment (approx. 300 meter with 3 curves / intersections). During those test trials no experimental instruction was given. Afterwards the instructor gave an estimation of the ease with which a subject would learn the task. Subsequently the subjects were assigned to one of the conditions (verbal instruction or augmented cues instruction). Subjects did not know what conditions the experiment comprised.

In both conditions subjects received the same four phases of a practice run followed by a test. During practice the instructor provided help (different depending on instructional condition). During the test no instruction was given (so that the tests were the same for both groups). After the test, the (subjective) instructor judgement was used to decide whether the trainee should continue to the next phase or repeat the same phase.

Each phase introduced some additional difficulties: Initially, the trainees were supposed to drive (at low speed) without shifting gears. No other cars drove around. In phase 2 other traffic was present. Traffic was removed again in phase three but gear shifting was introduced here. Finally in phase 4 trainees had to deal with other traffic while shifting gears.

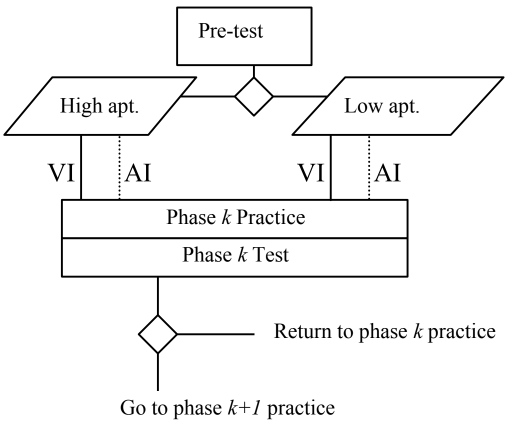


Figure 3: Experimental design. Diamonds represent a decision of the instructor. VI means 'Verbal Instruction', AI means 'Augmented Instruction'. After the pre-test, all subjects start in phase 1.

Hypotheses

It was hypothesized that differences in skill between groups of trainees would be reflected in the amount of instruction they received during *practice*, and in their performance during the *test* trials. Furthermore, we expected subjects that were classified as 'high-aptitude', would perform equally well regardless of the instructional condition they were assigned to. Low aptitude subjects however were expected to profit more from augmented instruction compared to verbal instruction.

Analyses

During the experiment different kinds of data were collected. Prior to the experiment a short questionnaire was administered with questions relating to traffic experience, gaming experience and knowledge of eight basic traffic signs that were to be encountered during the experiment. During practice the amount (and nature) of instructions provided by the instructor was stored for analysis. After the subsequent test the final instructor judgement was recorded (in this phase no instructions were given during driving). During both phases, driving performance was registered with regard to six dependent variables:

- Lateral position, i.e. the distance of the car to the center (white line) of the road.
- Standard deviation of the lateral position (stdevlp), for steadiness of position.
- Speed, in km/h. observing speed limits
- Standard deviation of speed (stdevsp), to register the variability of speed.
- Steering angle, amount of corrections of driving path.
- Standard deviation of steering angle (stdevsta). Smoothness of steering

Practice

Within each practice phase, a comparison of the (relative) amount of instruction between the experimental groups was made with an ANOVA. The differences over the experiment (i.e.) between the phases were not tested statistically because of task differences in each phase. Instead, a qualitative interpretation of these differences is provided.

Test

To evaluate the performance measures, in test phase 1 and 2 a MANOVA was used. An ANOVA was conducted in test phase 3. Because none of the subjects in the category ‘augmented / low’ participated in the fourth phase, no analyses were done for this test phase.

Results

Questionnaire

Of the 28 subjects, only one was licensed to drive a small motorcycle (<50 cc). None of the subjects had a theory license for car driving although 3 of them planned to take driving lessons within the next 1 to 6 months. Half of the subjects did not have any of such plans at all.

Only 13 subjects played computer games (four of them regularly). Car racing and ‘3D shoot 'em ups’ were both mentioned five times. None of the subjects used a steering wheel while playing.

The number of subjects that was able to name all eight traffic-signs correctly was 11. Twelve persons failed to name two or more signs. (The maximum number of missed or incorrectly named signs was four).

Practice

Only in the first phase the ANOVA yielded significant results, F (3,24) = 3.379, p = 0.03. A post hoc test (Tukey HSD) revealed that the AI low group received significantly more instructions than the AI high group. The differences between the AI-low and the other groups failed to reach significant values, although they were considerably far apart. This may be caused by the fact that Tukey's HSD is a rather conservative post-hoc test in that it offers a high amount of protection against the increased alpha error rate due to multiple post hoc comparisons. (For example, with the Newman-Keuls or Fisher LSD post-hoc test all three groups differ significantly from AI-low.)

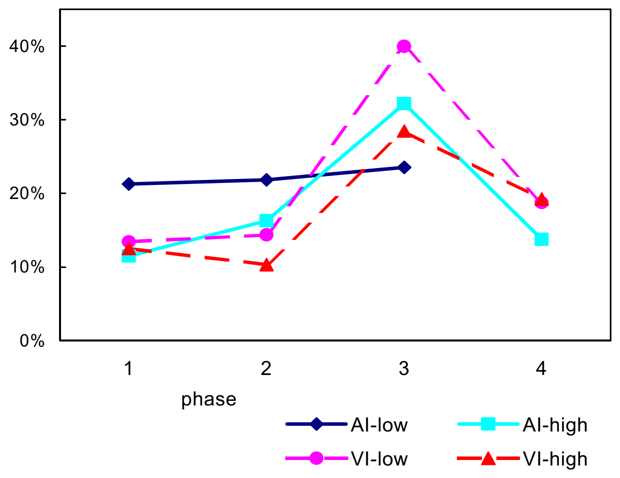


Figure 4: **Relative amount of instruction per practice phase for each condition group**

The ANOVAs for the other phases are not significant. Hence, no further tests were conducted.

Test

An ANOVA on the performance of trainees only showed a significant main effect between high and low aptitude trainees. That is, the high aptitude trainees performed better on all variables than the low aptitude trainees did. Remember that the difference between high and low was based on two short runs in the simulator. Whereas this might be seen as a result from instructor bias: the instructor both assigned trainees to the two categories and judged their performance, the objective data (performance measures) thus clearly support these findings.

With regard to the instruction conditions no significant differences were found on the performance measures in phases 1, 2, and 3. Phase 4 did not have enough data to compare the two instruction groups.

This same pattern was observed for performance on straight roads and in curves, and also for roads with a speed limit of 50 and 80km/h.

Discussion and Conclusions

The present experiment was the final one in a series of three. After the second experiment had been analyzed, it was hypothesized that a confounding variable (aptitude) possibly obscured the (positive) effects of the augmented instruction condition. This (third) experiment was set up to control for differences between subjects as much as possible by matching subjects on their aptitude for learning to drive.

For this means, the instructor was asked to give an estimate of each subject's aptitude based on their performance during a short driving test in the simulator[2]. Within that short period of observation, the instructor was able to assign each subject to either the high- or the low-aptitude group. Despite the subjective nature of this procedure, the experimental data (from the following sessions) clearly confirmed the instructor judgment. Yet it proved impossible to distinguish between subject aptitude based on (objective) demographic data, or knowledge of traffic rules as tested in a questionnaire that was also administered prior to the experiment. Contradictory as it may seem, this finding is in line with the results from studies that investigated a number of standard cognitive and psychomotor tests to give an indication of driving performance of neurological patients during the process of rehabilitation (e.g. Ball, Owsley, Sloane, Roenker & Bruni 1993; Heikkilä, 2000). A few examples of such tests are 'visual acuity, contrast sensitivity, eye health, visual memory, personality questionnaires, (choice) reaction time, and information processing tests. None of these has shown a high correlation with 'fitness for driving'. It seems that the best test for 'fitness for driving' is the driving task itself. The use of a driving simulator would allow for standardization of such a test in a safe environment.

With regard to the data it was observed that (just as in the second experiment), the amount of instruction delivered to the augmented group was larger than to the verbal group. On closer examination this is completely caused by the AI low group. This confirms that aptitude certainly is a variable that is capable of obscuring any effects of instruction. As none of the subjects in the augmented group complained about the (high) quantity of the instruction this may indicate that augmented instructions are less intrusive than verbal instructions and therefore may be easier to administer (or receive).

[2] Note that this was done prior to the experiment. Subjects had no driving experience and received only a short instruction on how to drive (the simulated car).

Augmented instruction may also be more adaptive than verbal instruction: If there is no need to provide additional instruction, it is used just as much as verbal instruction. This latter type of instruction seems to lack this flexibility. For low aptitude trainees then, it can be a solution to provide non-verbal (augmented) instructions because they seem to be friendlier. The exact (optimal) form of these augmented instruction is yet to be determined.

In previous experiments it was already seen that an error is not the sole criterion for a human instructor to give instruction or feedback. Only on about one out of six possible occasions, an error resulted in an instructor comment (Van Emmerik, De Jong & Van Rooij, 2000). One reason for this selective provision of feedback would be that the instructor would constantly be talking and the trainee would not have time to process the remarks / or direct attention to the (driving) task that is difficult anyway. Another reason may be that the instructor expects the trainee to learn recover from his own mistakes. Verbal instructions can be experienced as a criticism whereas augmented instructions can be seen as a hint leaving open the way to self recovery by the trainee. Therefore, they may be seen as especially helpful to low-aptitude trainees.

A final observation with regard to the augmented instructions was that the amount of AI for the low-aptitude group was smaller than for the other groups only in the third phase of the experiment. This contradicts the assumed flexibility of AI. After all, low aptitude trainees were expected to receive more instruction than their high aptitude counterparts. As a tentative post-hoc explanation it may be suggested that the instructor gave up on these low aptitude subjects. This seems to be supported by the fact that none of the trainees in this group reached the fourth phase.

No significant differences were found in performance in the test phases (even though there was a difference in the amount of instruction). The only difference that was found was in aptitude. Subjects categorized as low aptitude performed worse than high aptitude subjects. This observation, although of no importance to the experimental conditions, confirms the subjectivity of the instructor in judging performance during the experiments.

The interaction between aptitude and condition was never significant during the test phases suggesting that it does not matter what type of instruction (augmented or verbal) subjects received.

Although no definite conclusions can be drawn from these data, a trend was visible showing that low aptitude subjects performed slightly better with verbal instructions. This was something not expected as the augmented instructions were designed to be easily processed and therefore be beneficial in particular to those subjects that were having trouble to perform the task in the first place. Apart from that the (subjective) reactions of trainees were very positive with regard to the augmented cues.

Possibly, this surprising trend is a consequence of the differences between the twelve instructions. While some augmented instructions may have worked very good, others may have been difficult to interpret. This may have given the verbal instructions the overall advantage. Although these speculations could not be tested with this data, they may be an interesting topic for future research.

One thing this experiment may have cleared up though, is the question that remained open after previous experiments: It was suggested that the disappointing results of augmented instruction in that experiment could have been due to the coincidental assigning of low aptitude subjects to the augmented instruction group. This suggestion seems to be confirmed in the final experiment: After the subjects were matched on aptitude prior to assigning them to one of the experimental conditions, no significant differences were found between the instruction conditions (within the groups based on aptitude). The differences between these groups were highly significant however.

It could be true in general that the relatively small effect sizes of experimental manipulations with regard to instructional strategies are easily obscured by inter-individual differences. This is a problem that cannot be solved easily. Experiments of this type could never be done in a 'between subject' design because of the transfer between sessions. One solution probably is to use more subjects and to select them very strictly. Another possibility is to focus on a more detailed level of the driving task. For example, a researcher could pick out one particular instruction to compare different forms of. Both approaches are very laborious. An additional disadvantage of the latter approach is that it would require a task that is more abstract than the current driving task. This would also reduce the validity of the experimental environment and restrict the range of the conclusions.

References

Ball, K., Owsley, C., Sloane, M. E., Roenker, D. L., & Bruni, J. R. (1993). Visual Attention Problems as a Predictor of Vehicle Crashes in Older Drivers. *Investigative Ophtalmology & Visual Science, 34*(11), 3110-3123.

Cook, C. A., O'Shea, A. M., Young, A. L., & Stedmon, A. W. (1999). Helpful feedback? Deriving guidelines for advanced training systems. In *Proceedings of the Human Factors and Ergonomics Society 43rd annual meeting* (pp. 1133-1137).

Emmerik, M. L. v., Rooij, J. C. G. M. v., & Jong, T. d. (2000). *Exploration of Instruction in a Driving Simulator.* (Report No. TM-00-B008). Soesterberg, The Netherlands: TNO-Human Factors.

Heikkilä, V. M. (2000). Relationship of laboratory and on-road tests for driving-school students and experienced drivers. *Perceptual Motor Skills, 90*, 227-235.

Korteling, J.E. & Sluimer, R. R. (1999). *A critical review of validation methods for man-in-the-loop simulators.* (Report No. TM-99-A023). Soesterberg: TNO-Human Factors.

Lintern, G., Thomley-Yates, K. E., Nelson, B. E., & Roscoe, S. N. (1987). Content, variety, and augmentation of simulated visual scenes for teaching air-to-ground attack. *Human Factors, 29*(1), 45-59.

Lintern, G. & Koonce, J. M. (1992). Visual augmentation and scene detail effects in flight training. *The International Journal of Aviation Psychology*, 2(4):281-301.

Lintern, G.; Roscoe, S. N., and Sivier, J. E. (1990). Display principles, control dynamics, and environmental factors in pilot training and transfer. *Human Factors*, 32(3):299-317.

O'Shea, A. M., Cook, C. A., & Young, A. L. (1999). The use of augmented reality in the provision of training feedback. In *International Training and Education Conference (ITEC)* The Hague.

Roscoe, S. N. (1991). Simulator qualification: Just as phony as it can be. *The International Journal of Aviation Psychology*, 1(4):335-339.

Schneider, W. (1985). Training High-Performance Skills: Fallacies and Guidelines. *Human Factors*, 27:285-300.

Summala, H., Lamble, D., & Laakso, M. (1998). Driving Experience and Perception of the Lead Car's Braking when looking at In-Car Targets. *Accident Analysis and Prevention,* 30(4): 401-407.

Young, A. L., Stedmon, A. W., & Cook, C. A. (1999). The potential of augmented reality technology for training support systems. In *People in Control* ,Bath, UK, June 1999.

Using Embedded Pedagogical Tools for Optimizing Within Visual Range Combat Training – the Initial Experience

Staffan Nählinder, Jonathan Borgvall, Erland Svensson

FOI – Swedish Defence Research Agency
P.O. Box 1165
SE-581 11 LINKÖPING, Sweden
staffan.nahlinder@foi.se

Abstract

Many of today's combat flight simulators have a high level of realism regarding most important levels of fidelity. Unfortunately, most often, they suffer regarding explicit training tools or "pedagogical tools". Even though the computer evolution has made it possible, very few, if any, embedded pedagogical tools are available in the Swedish military simulator training facilities of today. In a simulated environment, it is possible to embed tools and visualizations that effectively can enhance training efficiency in ways not possible in real world situations. When developing a new simulator, it might in fact be a very effective strategy to focus (some) resources towards pedagogical aspects rather than just continue pursuing the impossible goal of achieving perfect realism. The authors believe that by relaxing the realism - and instead stressing the pedagogical possibilities - a much more effective training environment can be accomplished. In the development of the present "WVR Demonstrator" (Within Visual Range), such a strategy has been chosen. The simulator is a second version of the "WVR Illustrator" (presented at SAWMAS 2002). It is much improved and now has several embedded pedagogical tools for training. It is now possible for the pilots to see a visualization of several important combat parameters in real time in order to maximize the usefulness and training efficiency of the simulator. During a combat situation, these tools can be used individually or together according to the preference of the pilot and instructor. In the present report, a study of the training effects of such tools is presented. Swedish military pilots have answered several questions regarding (among other things) the usefulness of the pedagogical tools in the "WVR Demonstrator". Preliminary results from the study suggest that the pedagogical tools are most probably very useful, at least in early stages of combat flight training. Pilots in the study believe that, if used correctly, the embedded pedagogical tools can make WVR-combat training more effective. The results are yet preliminary; however, they clearly show an effective step forward in the development of training simulators.

Introduction

Debriefing as a method for learning and improving has historical roots (Pearson & Smith, 1986) and could be closer described as a discussion performed after a mission with the purpose of improving future performance. One definition is that debriefing is the process of learning through the reflection of an experience (Thatcher, 1986; Pearson & Smith, 1986; Lederman & Stewart, 1986). Debriefings specifically used in training situations are often referred to as after action-reviews (Downs, Johnson & Fallesen, 1987).

The normal debriefing procedure in combat flight training today is that the student pilot is debriefed by the instructor after a flight. The obvious problem with this is that both of them have to remember and recall the situations and manoeuvres, often after several air-to-air contacts. Even if they could remember the situations correctly it is impossible to repeat exactly the same situation at a later point in time (Danielsson, Svensson & Jenvald, 2002). That is, the learning on how to handle a specific situation might be restricted to the feedback given by an instructor based on what is remembered after very few encounters of that particular situation. This is an important matter since debriefing is an important part of experience-based learning processes (Lederman, 1992). Research by Stammers in 1983 support this idea, since he isolated the provision of extrinsic feedback as a key training variable. However, if appropriate feedback for learning of the task is not present in the real system, some form of simulator practice is desirable (Stammers, 1983). Any task that one wants to master should be repeated many, many times in order to optimize the learning of the task (Ericsson and Simon, 1993; Fisk and Lloyd, 1988). Research suggest that experience play a key role in performance among fighter pilots (Angelborg-Thanderz, 1990). According to Stedmon and Stone (2001), virtual environments offer improved effectiveness, reduced reliance and enhanced training methods over and above traditional methods. This possibility is one important reason for integrating comprehensive pedagogical tools in the Demonstrator.

The pedagogical problems discussed above and the increasing potential of new technology has raised the issue of developing a high-fidelity simulator for air-to-air combat training with embedded extensive pedagogical tools for feedback during and after combat (Svensson, Nählinder, Danielsson & Jenvald, 2002). During the autumn of 2001, the development of a first prototype started off in cooperation between the Swedish Armed Forces Headquarters (HKV) and the Department for Man-System Interaction (MSI) at the Swedish Defence Research Agency (FOI). The system was called the Illustrator and the main purpose was to develop the prototype in a short time and to a low cost. No pedagogical tools were integrated at this stage. In March 2002 the Illustrator was delivered to FOI-MSI for evaluation using fighter pilots from the Swedish Air Force, who identified some major weaknesses of the system. This simulator evaluation was presented at SAWMAS 2002 (Svensson, Nählinder, Danielsson & Jenvald, 2002). However, the results of the evaluation also indicated a great potential for this type of system (Borgvall, Nählinder & Andersson, 2002) and a decision was made to continue the project by developing a more advanced prototype. The extensive goal of the project was yet to develop a simulator prototype with comprehensive pedagogical tools for WVR-combat fighter pilot training (Danielsson, Svensson & Jenvald, 2002; Svensson, Nählinder, Danielsson & Jenvald, 2002). The guidelines that the evaluation of the Illustrator generated were used during the specification of the new prototype. The second prototype is called the Demonstrator and it will complement the present simulation facilities in the Swedish Air Force.

To conclude the discussion of the pedagogical tools, there are a range of military studies that have clearly demonstrated that computer-based training increased performance improvement, even in comparison with hands-on training with real equipment (e.g. Stedmon & Stone, 2001; Stone, 2001).

WVR Combat

The WVR-Demonstrator is a flight simulator specifically built for within-visual range (WVR) combat training.

Figure 1. A simulator cabin in the WVR Demonstrator

The Swedish Air Force has several simulators for fighter pilot training. Concerning the visual presentation, those systems are optimized for beyond visual range combat (BVR-combat) simulation. During this type of combat, the fighting aircrafts have no or at least very limited visual contact. The activities related to BVR-combat are therefore primarily based on instrument flying. This means that the demands on the visual presentation of the surrounding environment are restricted in contrast to within visual range combat (WVR-combat). During a WVR-combat situation (often referred to as dogfight), the fighting aircrafts have a maximum distance of approximately 15 kilometres, depending on weather conditions. During WVR-combat, the combatants try to maintain visual contact to be able to perform the smartest manoeuvres in order to reach a location where it is possible to fire at the enemy, or at least make sure that the enemy can not fire back. Most often, the best location is to end up directly behind the enemy aircraft. The most optimal position depends on what weapons are available and what weapons the enemy carries. One of the challenges when learning how to master WVR-combat, is to understand the importance of having a superior energy level. The energy level is the sum of energy from the velocity and the energy from the altitude. That is with high altitude and high speed you are more likely to win than with low altitude and low speed. The geometrical relationships between the aircraft must be understood as well as the relationship between manoeuvring and maintaining speed.

In WVR-combat the instruments are almost exclusively used for controlling the status of the own aircraft (height, speed, altitude). In many simulators, the visual presentation is static in the sense that the operators see a limited field of what is in front of the aircraft and the view does not change according to the operators' head movements. The exceptions are domes and head-mounted displays with head-tracking systems. For BVR-combat simulation, this is not believed to be necessary, but for WVR-combat it is. In WVR-combat, the operator has to be able to follow the enemy visually in the simulated environment. Domes and HMD´s offer this possibility, with HMD´s as the cheaper alternative. However, one major question is whether the HMD´s available today are of sufficient quality for high fidelity air-to-air combat simulation.

In the Illustrator evaluation, one of the key results was that the HMD's used in that simulator had a much too narrow field-of-view. The fighter pilots that tested the Illustrator all agreed that the limited field-of-view was the single most limiting feature regarding its use as a training simulation. The importance of a large field-of-view is further discussed in Burki-Cohen, Tiauw & Longridge (2001).

The WVR Demonstrator

The WVR Demonstrator was built in 2003 and consist of two pilot stations (cabins), see Figure 1, and a instructor station. The instructor can manipulate the scenario and can also control the pedagogical tools. In the pilot stations, the head down displays, the flight instruments, are presented on a computer screen inside the cabin. The world is displayed inside head mounted displays which offer a quite large field of view, see Figure 2. The HMD's used in the Demonstrator has a much larger field-of-view than the ones used in the Illustrator. The image in the head mounted displays are updated according to the pilots head movements, that is, if the pilot turns his or her head towards the left, the HMD will also display the view looking out to the left of the aircraft. The head movements are measured in six dimensions, so not only the head direction is detected but also the head position. As a total, the HMDs generate a very compelling out-of-the-window feeling. One can almost lean out of the cockpit and look below the aircraft!

The pilots communicate with the instructor through a headset.

Figure 2. Head mounted display in the WVR Demonstrator

Normally, the scenario is set up that the two pilot stations will compete against each other trying to manoeuvre into a favourable position behind the opponent aircraft. However, it is also possible to join forces and to fight simulated enemies together. Each pilot station can be run separately for instance while training advanced flying skills.

Pedagogical tools in the Demonstrator

Most simulators today have several imbedded pedagogical tools, such as the possibility to freeze the scenario, or to manipulate environmental values, such as visibility, wind and the time of day. Many simulators also support (at least limited) after-action-review. However, most often, the range of pedagogical tools is limited. However, today with the overwhelming and constantly improving power of computers, there is much more that can be done!

Spatial tools

In the WVR "Demonstrator", it is possible to enhance the visual understanding of the geometrical relationships between the aircraft by seeing the scene from any position of the simulated world. For instance, if one has trouble understanding the opponents' point of view, one can simply freeze the running combat and then move ones eyes placing them inside the enemies' aircraft. This way, the pilot in the first aircraft can see exactly what the opponent sees and understand his or her situational awareness. One can also place ones eyes at any other position in the simulated environment. For instance it is possible to see the scene from a God's eye position in the sky! The pilot can control the position of the eyes from his cockpit.

To further enhance the geometrical understanding of the aircraft, it is possible to add 3D tapes behind each aircraft showing the latest path of the aircraft, see Figure 3. The visual appearance of the aircraft can be improved by scaling up the aircraft.

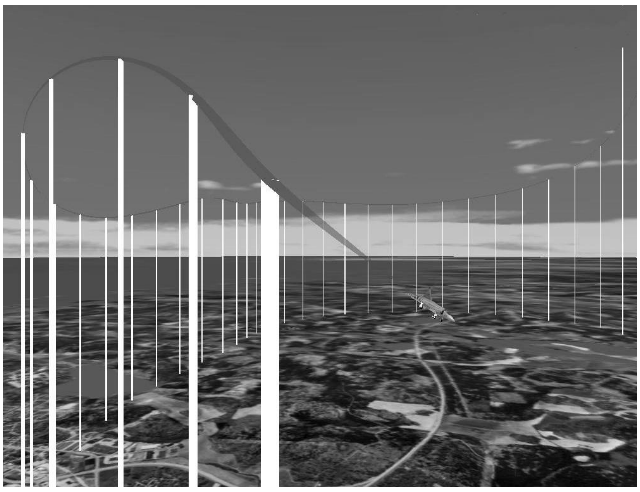


Figure 3. 3D bands showing the aircrafts path.

It is also possible to switch aircraft with the opponent in mid flight without leaving the cabins. Or both pilots can be placed in the same aircraft (however only one at a time can handle the controls).

Time tools

There is a feature to rewind and redo a situation. By simply moving back the time, the two aircraft can redo the exact same situation again. Further either none, one or both pilots can choose to replay whatever happened last time, or to do a fly-out from the same initial condition but fly differently. If replay is chosen, the aircraft will fly the exact same path as last time, and the pilot can literally lean back and enjoy the ride! After several different fly outs, it is possible to display all the flight paths in the air, creating a sun feather effect displaying each of the flights overlaying each other. This supports the pedagogical effect of directly comparing the outcomes of different flight strategies.

It is also possible to further manipulate time. For instance, time can be played backwards, or forwards at either normal speed, below normal speed or above normal speed. Some studies show that by training at above-speed simulation, real flight then becomes less stressful than if the training was done at normal speed. On the other hand, training at below normal speed increases the feeling of situational awareness and thus might also facilitate in the learning process.

Combat analyses

Besides the embedded pedagogical tools for manipulating space and time, the Demonstrator also includes possibilities of real-time analyses of important combat parameters. For instance, a direct comparison of aircraft energies is possible using graphs that display the energy history of the two aircraft. This makes it intuitively easy to see what aircraft can maintain an energy advantage over his or her opponent.

Other analysis tools include comparisons of speeds, altitudes and energy loss due to extensive maneuvering. Several other analysis tools are also available especially concerning analysis of relative positions between the aircraft.

These combat analyses graphs all help to understand who won the combat and more importantly why.

Evaluation

To evaluate the usefulness of the pedagogical tools in the WVR Demonstrator, some initial qualitative data regarding the potential future training value of the pedagogical tools was collected through the use of questionnaires. When collecting such data it is important to use the future users of the system to answer the questionnaires. Otherwise it will be difficult to generalize the results to the relevant population.

As of today, only limited data has been collected regarding the end users ratings of the pedagogical features mentioned above. Four experienced fighter pilots have scrutinized the tools and subjectively evaluated the perceived usefulness of each tool. They all agree that the tools will be very useful, especially in early combat flight training.

In the short time future, a more extensive data collection will be done asking several more expert fighter pilots about the usefulness of the pedagogical tools.

The best way to evaluate the usefulness of the pedagogical tools would be to perform a test of the transfer-of-training effect. In such a study, one could take a group of relatively new pilot students, and divide the group into two. Both groups would train their combat flight skills in the Demonstrator, but only the first group would be allowed to use the pedagogical tools. After extensive training, one would let each student from one group compete against a student from the other group in a real-flight combat situation. If the students from the first group would win significantly more often than the second group, the pedagogical tools would most probably be useful!

Obviously, such studies can rarely be done. There are many issues to account for before it could possibly be done. Therefore, with the present Demonstrator, subjective questionnaire data will be collected instead, focusing on the perceived usefulness of the pedagogical tools at hand. When preparing for such a study, one must always remember what the purpose of the simulator is. The key issue must always be operative performance! It doesn't matter if it looks good or feels good: if it doesn't improve performance in the real world, it's of no use. Operative performance is quite difficult to measure. Often a particular behaviour can be affective even though it is difficult to quantify. For instance in an intercept mission, a good performance could be to scare off the enemy even without firing missiles. However, in other situations the *only* good performance is to shoot down the enemy before he or she shoots down you!

Conclusions

The main study on the usefulness of the WVR Demonstrator is yet to come. Swedish Air Force pilots will be used to answer questionnaires regarding many different aspects of the simulator. Primarily, the usefulness of the pedagogical tools will be analyzed.

Besides evaluating the pedagogical tools, a fidelity evaluation (Roscoe, Wilkinson & VanderVliet, 2001) will also be made similar to the one performed on the Illustrator. User acceptance will be measured concerning different aspects of the simulator (see Bell & Waag's, 1998 regarding the importance of user acceptance; Salas, Bowers and Rhodenizer, 1998).

The main study is scheduled to be performed in the beginning of 2004.

So far however, it seems that the pedagogical tools can in deed be very useful, especially when it comes to understanding the importance of maintaining superior geometrical location relative to your opponent. Also, the usefulness of the pedagogical tools might be largest when teaching fresh pilot students WVR combat. Initially, many pilot students have a hard time understanding the geometrical relationships between two fighting aircraft and the importance of avoiding certain situations. With the tools described above, this understanding can be achieved much earlier in the training, and hence make combat flight training more effective.

The results so far seem to show that the implementation of pedagogical tools in the simulator might make training more effective. Future simulator development projects should consider these findings and thoroughly investigate what tools to embed. The pedagogical tools described above only start to illustrate what possibilities lay in training simulators. Through extensive research and cognitive task analysis it should be possible to find similar useful pedagogical tools in any training situation, not only in flight simulators.

Since the study has only just begun, the results should be interpreted with great care. However, the authors believe that the pedagogical tools are a very useful way forward and an important step in future training simulator development projects!

References

Angelborg-Thanderz, M. (1990). *Military Flight Training at a Reasonable Price and Risk*. Swedish National Defense Research Establishment report C 50083-5.1.

Bell, H. H., & Waag, W. I. (1998). Evaluating the Effectiveness of Flight Simulators for Training Combat Skills: A Review. *The International Journal of Aviation* Psychology, 8(3), 223-242.

Borgvall, J., Nählinder, S., & Andersson, J. (2002). *WVR-Illustrator Evaluation: Using Pilot Expertise for Future Development* (Scientific FOI-R--0710--SE). Linköping: Department for Man-System Interaction.

Borgvall, J., Nählinder, S., & Andersson, J. (2003, June 2-4). *Using Pilot Expertise for the Evaluation of a Within-Visual-Range Air-to-Air Combat Simulator Prototype*. Paper presented at the The 22nd European Conference on Human Decision Making and Manual Control, Linköping.

Burki-Cohen, J., Tiauw, G., & Longridge, T. (2001). *Flight Simulator Fidelity Considerations for Total Airline Pilot Training and Evaluation* (A01-37337). Montreal, Canada: AIAA.

Danielsson, B., Svensson, E., & Jenvald, J. (2002). *Pedagogical Tools for Within Visual Range (WVR) Fighter Pilot Training* (Reprint FOI-S--0453--SE). Linköping: Division of Command and Control.

Downs, C. W., Johnson, K. M. and Fallesen, J. J. (1987). *Analysis of Feedback in After Action-Reviews*. AD A188 336. U.S. Army Research Institute for Behavioural and Social Sciences, Alexandria, Virginia, USA.

Ericsson, K. A. & Simon, H. A. (1993). *Protocol Analysis* (Second Ed.). Cambridge: MIT Press.

Fisk, A. D. & Lloyd, S. J. (1988). *The Role of Stimulus-to-Rule Consistency in Learning Rapid Application of Spatial Rules*. Human Factors, 30, 35-49.

Lederman, L. C. and Stewart, L. P. (1986). *Instructional Manual for THE MARBLE COMPANY: A Simulation Board Game*. New Brunswick, New Yersey: SCILS.

Lederman, L. C. (1992). *Debriefing: Toward a Systematic Assessment of Theory and Practice*. Simulation & Gaming, 23(2), 145-160.

Pearson, M. and Smith, D. (1986). *Debriefing in Experience-Based Learning.* Simulations/Games for Learning, 16(4), 155-172.

Roscoe, M. F., Wilkinson, C. H., & VanderVliet, G. (2001). The Use of ADS-33D *Useable Cue Environment Techniques for Defining Minimum Visual Fidelity Requirements* (A01-37380). Montreal: AIAA.

Salas, E., Bowers, C. A., & Rhodenizer, L. (1998). It Is Not How Much You Have but How You Use It: Toward a Rational Use of Simulation to Support Aviation Training. *The International Journal of Aviation Psychology*, 8(3), 197-208.

Stammers, R. B. (1983). Simulators for training. In T. O. Kualseth (Ed.), *Ergonomics of Workstation Design* (pp. 229-242). London: Butterworths.

Stedmon, A. W., & Stone, R. J. (2001). Re-viewing reality: human factors of synthetic training environments. *International Journal of Human-Computer Studies*, (55), 675-698.

Stone, R. (2001). Virtual Reality for Interactive Training: An Industrial Practitioner´s Viewpoint. International *Journal of Human-Computer Studies*, (55), 699-711.

Svensson, E., Nählinder, S., Danielsson, B., & Jenvald, J. (2002, October 30-31). *Pedagogical Tools for Within Visual Range (WVR) Fighter Pilot Training - Another Step Forward.* Paper presented at the Swedish American Workshop on Modelling and Simulation SAWMAS-2002, Orlando, FL, USA.

Thatcher, D. (1986). Promoting Learning Through Games and Simulation. *Simulation/Games for Learning*, 16(4), 144-154.

Supportive Behavior, Frustration, and Mental Demand: Implications for Perceived and Actual Team Performance

Kia Asberg, Clint Bowers, and Eduardo Salas

University of Central Florida
Team Performance Laboratory
12424 Research Parkway
Orlando, FL 32816
kiaasberg@hotmail.com

Abstract

Investigations of behaviors and processes taking place in complex team decision-making environments are important to assure optimal performance and prevent negative effects of stress in individuals and teams. Because various types of stress, such as increased workload or time pressure, often lead to decrements in team performance in the absence of protective factors, the present study explored social support as a moderator in the stress-performance relationship. The present study hypothesized that the frequency of socially supportive exchanges between members would predict lower levels of frustration and mental demand, as well as be related to a better overall team performance on a 30-minute computer-based task. A secondary hypothesis stated that frustration levels and perceived mental demand (workload) of individual members would be related to lower team performance. Limitations of this study, suggestions for future research, and implications for team training will also be discussed.

Introduction

Recent trends and structural changes in modern tasks have imposed additional demands (both mental and physical) that may be difficult, if not impossible, for individuals to handle alone (e.g., Salas, Dickinson, Converse, & Tannenbaum, 1992). Salas et al. (1992) also note that technical developments and global competition has made the ability of individuals to function together as a productive team increasingly important. Furthermore, over the past few decades, as the body of research investigating complex team decision-making environments has grown, the relationship between stress and team processes has become more clearly defined. In general, the team literature indicates that various types of stress, such as increased workload or time pressure, often lead to decrements in performance in the absence of protective factors (e.g., Bowers, Weaver, and Morgan, 1996; Glaser, Tatum, Nebeker, Sorenson, and Aiello, 1999; Urban, Weaver, Bowers, and Rhodenizer, 1996). In recent years, investigators have highlighted the importance of optimizing team performance in areas of aviation, military and emergency operations, as well as other stressful and complex environments where teams are utilized (Weaver, Bowers, Salas, 2001).

Team Definition and Interdependence. To describe "team" in the present study, we employ the common definition proposed by Salas, Dickinson, Converse, and Tannenbaum (1992), where a team refers to two or more individuals who interdependently are working toward a common goal. Teams most often investigated in this line of research involve those with requirements of coordination and a high degree of interdependence (Weaver et al., 2001). In addition to the obvious differences, Weaver et al. (2001) note that stress may affect teams differently compared to individuals in that both interactions between team members *and* individual performance must be maintained during a task. The type of team interdependence in focus of the present investigation can best be described as "sequential interdependence" (Saavedra, Earley, & Van Dyne, 1993). Under sequential interdependence, Saavedra et al. (1993) proposes that "one group member must act before another can act" (p. 62). Furthermore, team members have "different roles and perform different parts of the task in prescribed order" (Saavedra et al., 1993, p. 62-63).

Team Stress. Extending the Lazarus et al. (1984) definition of stress, *team stress* involves an interaction between the team and its environment where perceived demands outweigh resources (e.g., Weaver et al., 2001). Furthermore, Griffith (1997) investigated stress, strain, and group disintegration in 9013 U.S. Army soldiers, and found that all three factors added significant variance to perceived individual and group performance. Although results regarding the effects of various types of stress on team performance are somewhat inconclusive, the literature appears to agree that adaptability plays a key role (e.g., Serfaty, Entin, & Volpe, 1993).

Temporal Demand. As noted by Weaver et al. (2001), time pressure (temporal demand) is the one stressor that has received most of the attention in the team performance literature. Most studies have indicated that time pressure has a detrimental effect on team performance, however, the relationship is not always linear (e.g., Serfaty, Entin, & Volpe, 1993). The present study investigated variables related to team performance during a 30-minute task where participants were given feedback regarding "time remaining for completion of mission objectives".

Hypotheses. The present study hypothesized that the frequency of socially supportive verbal exchanges between members would predict lower levels of frustration and perceived mental demand, as well as be related to a better overall team performance on a 30-minute computer-based task designed to simulate a reconnaissance mission. A secondary hypothesis stated that higher frustration levels and higher perceived mental demand of individual members would be related to lower overall team performance, both actual and perceived. Our conceptual framework and main study variables will be described in more detail below.

Stress and Performance

As previously discussed, effects of different types of stress or stressors on individual performance has received extensive attention over the past decades (e.g., Weaver, Bowers, & Salas, 2001). Adopted from Lazarus and Folkman (1984), the most commonly used definition of stress in the area of human performance suggest that stress is an interaction between an event and an individual's perception or interpretation of the particular event (Weaver et al, 2001; Driskell and Salas, 1996). The ways in which stressors, such as time pressure, affects team performance, however, is a relatively new area of investigation (Weaver et al., 2001). Other constructs associated with stress in the literature include, but are not limited to, strain, mental load, burnout, and fatigue (e.g., Tepas & Price, 2001). From a more global perspective, it appears that stress and stress prevention are widely recognized as important issues in terms of safety and health of workers (Kompier and Cooper, 1999).

Social Support

One variable that has garnered interest as a potential moderator in the stress - performance relationship is social support. Many different definitions of social support exist, but Searle, Bright, and Bochner (2001) operationalize the concept as "a variety of resources that assist an individual in their work and daily life, such as task-relevant information or praise" (p. 328). In a recent study, Brotheridge (2001) investigated co-worker support, workload, and emotional exhaustion in 680 Canadian government employees. Findings suggested that coping resources (i.e., social support) protected workers from experiencing strain regardless of their pre-existing levels of stress. Furthermore, several studies have suggested that higher levels of social support is related to less burnout (e.g., Fielding & Weaver, 1994) and higher satisfaction among workers (e.g., Decker, 1985). Previous studies have viewed social support as a subcomponent of team cohesion, a variable that has been identified as "an important ingredient for combat effectiveness and performance" (Griffith, 1997, p. 1489). Griffith (1997) further divided the cohesion/support variable into team task support and peer emotional support. In social and organizational psychology, researchers have also found a relationship between group cohesion and productivity (e.g., Evans & Dion, 1991).

Frustration

By definition, novel performance environments, such as the one encountered by participants in the present study, entail unfamiliar aspects (e.g., Marks, 1999). Such environments are also believed to be less predictable and more challenging (Marks, 1999). When expectations for individual or team performance are not met, a person is likely to experience dissatisfaction or frustration. Reber (1995) suggests that the technical use of the term frustration in psychology generally refers to one (or both) of the following two definitions: 1) "the act of blocking, interfering with or disrupting behaviour that is directed toward some goal" (p. 301) (e.g., the behaviour may be anything from overt, physical movement to covert, cognitive process), or 2) "the emotional state assumed to result from the act in the first definition" (p.301). Furthermore, Reber (1995) states that 'it is typically assumed that this emotional state has motivational properties that produce behaviour designed to bypass or surmount the block." (p. 301). In lieu of this definition of frustration, it makes sense to believe that the concept will be related to team performance.

Mental Demand

Mental demand and stress are closely related in that both concepts focus on the balance between demands and resources, however, mental demand (also referred to as mental load) specifically targets the task and the internal resources needed to perform it (Gaillard, 2001). Stress on the other hand describes "the fit between the person and the environment" (Gaillard, 2001, p. 628). Although different definitions exist, mental demand most often refers to the relationship or ratio between an operator's capacity and the capacity required by the task for execution (e.g., Hancock, 1987).

Method

Participants. A total of 24 undergraduate students at the University of Central Florida who volunteered to participate in the experiment were randomly assigned to one of two roles (computer navigator or map reader), and formed 12 two-person distributed teams. Although not required for participation, all team members had some experience with playing computer games, however, none of the participants had encountered the particular game or task at hand. Following a training session where participants were given the opportunity to become familiar with the key-board commands, specific standardized task instructions were given until teams displayed competency (as indicated by completion of a trial mission). Participants were also informed about the overall performance objective, and the time-limit for completing the task.

Measures and Design. Verbal communication (audio) and task performance (video) was recorded for each team, and participants completed the NASA-TLX (TLX; Hart and Staveland, 1988) as a measure of their perceived stress on six domains, including perceived mental demand, temporal demand, effort, physical demand, frustration, and performance. Bowers, Brown, and Morgan (1997) suggested that the use of the NASA-TLX, a subjective measure of workload, works well in team settings. The NASA-TLX has shown good psychometric properties in previous studies of workload (reported test-retest reliability = .83). In addition to the self-report measure of workload dimensions, a behaviours coding system was used to assess socially supportive exchanges, as well as negative exchanges, between team members during completion of the task. Specifically, with a modified version of Operation Flashpoint, a Commercial Off The Shelf (COTS) computer game as a test-bed, the team task involved verification of targets in the virtual environment. Separated by a divider in the laboratory, each distributed team had access to physical maps and written clues pertaining to target locations, but members were unable to provide each other with visual cues. In addition, team members were instructed to only share their specific information verbally, although no other restrictions on communication were set a priori. The simulated task can best be described as an intelligence based mission where verbal communication is the key to successful performance. Furthermore, discrete behavioural coding categories for communication between team members included encouragement, positive and negative task-related feedback, etc., and each team's communication was coded by two trained research assistants. Next, relationships between social support frequency and variables such as Perceived Frustration, Perceived Performance, and Mental Demand (as indicated by scores on the NASA-TLX subscales) were investigated. Furthermore, Actual Team Performance on the computer-based task was calculated based on the number of targets a team was able to locate in the virtual environment during a "scavenger-hunt" type mission. Specifically, team performance was operationalized as the number of targets (out of 24 possible) identified during the 30-minute mission. For example, each team's main objective was to verify the truthfulness of statements such as: "Terrorists have hidden a computer on the second floor of building 15" and "A civilian is hiding in the house across from the south fountain".

Results

Results from initial analyses of the data indicated a significant negative correlation between team members' level of frustration and their actual team performance score (p=.007). A positive correlation was found between frustration levels and perceived team performance (p=.043), with higher score on the perceived performance scale indicating worse performance. Frustration levels and mental demand variables were also positively correlated to a significant degree (p=.003). Perceived individual performance and actual team performance were negatively correlated (p=.007), again, note that higher scores on the individual performance sub-scale of the NASA-TLX indicates worse perceived performance. Finally, perceived team performance and perceived individual performance were significantly correlated at the .05 alpha level (p=.001). Contrary to our stated hypothesis, no significant relationship between the frequency of socially supportive behaviors and study variables was found. However, it is important to note that this absence of an effect may be due to the limited range of supportive behaviors expressed by team members in general. On average, teams exchanged 2.08 supportive statements (SD=1.692) and .50 negative (unsupportive) statements (SD=.722) per 30 minute session. Moreover, temporal demand, physical demand, and effort, as indicated by the NASA-TLX, did not show significant relationships with any variables in the present study. The inherent difficulty of the task was illuminated by the fact that only one team was able to identify all 24 target statements within the allotted time, while the average team performance score was 13.84, or approximately half of possible points. Further analyses of the data (i.e., multiple regression) will be conducted to investigate more refined relationships between study variables, as well as to shed light on the stress-performance relationship among sequential, distributed teams.

Summary

Previous research has suggested that supportive interactions among team members may buffer against stress effects and protect the team against performance decrements (e.g., Evans et al., 1991; Griffith, 1997). Although several definitions of stress related concepts, as well as moderators, have been proposed, enough evidence has been accumulated to support a relationship between task demand and performance, both perceived and actual. For example, various types of stress (e.g., unpredictability/ambiguity of a novel team-task and time pressure) may be related to increases in perceived mental demand and frustration, both of which has been found to affect performance negatively. Although the task at hand was believed to be inherently "stressful" due to the imposed variables described above, we hypothesized that receipt of socially supportive behaviours would ameliorate perceived mental demand and frustration, thereby protecting teams against performance decrements. Contrary to our beliefs, results of the present investigation indicate that receipt of supportive behaviors was not correlated with any of the study variables, such as performance, perceived mental demand, and frustration. However, as mentioned, the limited range (floor effect) of supportive behaviors is likely to have contributed to these findings. For future research, we suggest that social support is investigated as a manipulated independent variable, which will allow for group differences to be explored. In terms of team training, the present study addressed the importance of identifying variables that may moderate participants' experience of mental demand and frustration. Mental demand and frustration were both found to be related to team performance, and although social support demonstrated no significant correlation with these variables in this investigation, we argue for its potential as a stress reducer based on previous research.

References

Bowers, C., Braun, Morgan, B. (1997). Team workload: Its meaning and measurement. In R. Swezey & E. Salas (Eds.) *Team performance assessment and measurement: Theory, methods, and applications* (pp. 85-108). Mahwah, NJ: Erlbaum.

Bowers, C., Weaver, G., & Morgan, B. (1996). Moderating the performance effects of stressors. In J. Driskell & E. Salas (Eds.) *Stress and Human Performance* (pp.163-192). Mahwah, NJ: Erlbaum.

Brotheridge, C.M. (2001). A comparison of alternative models of coping: Identifying relationships among coworker support, workload, and emotional exhaustion in the workplace. *International Journal of Stress Management, 8(1)*, 1-14.

Cohen, S., & Willis, T.A. (1985). Stress, social support and the buffering hypothesis. *Psychological Bulletin, 98,* 310-357.

Decker, F.H. (1985). Socialization and interpersonal environment in nurses' affective reactions to work. *Social Science and Medicine, 20,* 499-509.

Driskell, J.E., & Salas, E. (1996). Stress and human performance (pp. 163-192). New York, NY: Erlbaum.

Evans, C.R., & Dion, K.L. (1991). Group cohesion and performance: A meta-analysis. *Small Group Research, 22(2),* 175-186.

Fielding, J., & Weaver, S.M. (1994). A comparison of hospital and community mental health nurses: Perceptions of their work environment and mental health. *Journal of Advanced Nursing, 19,* 1196-1204.

Glaser, D., Tatum, C., Nebeker, D., Sorenson, R., & Aiello, J. (1999). Workload and social support: Effects on performance and stress. *Human Performance, 12(2),* 155-176.

Griffith, J. (1997). Test of a model incorporating stress, strain, and disintegration in the cohesion-performance relation. *Journal of Applied Social Psychology, 27(17),* 1489-1526.

Hancock, P.A. (1987). *Human Factors Psychology.* Amsterdam: North-Holland.

Hart, S.G., & Staveland, L.E. (1988). Development of NASA-TLX: Task Load Index. In P.A. Hancock & N. Meshkati (Eds.), *Human mental workload,* pp. 139-183. Amsterdam:Elsevier North-Holland.

Keinan, J., & Friedman, N. (1996). Training effective performance under stress: Queries, dilemmas, and possible solutions. In J. Driskell & E. Salas (Eds.), *Stress and Human Performance*, pp. 257-277. Mahwah, NJ: Erlbaum.

Kompier, L., & Cooper, C.L. (Eds.) *Preventing stress, improving productivity: European case studies in the workplace.* London and New York: Routledge.

Lazarus, R.S., & Folkman, S. (1984). Stress, appraisal, and coping. New York, NY: Springer.

Marks, M.A. (1999). A test of the impact of collective efficacy in routine and novel performance environments. *Human Performance, 12(3/4),* 295-309.

Reber, A.S. (1995). Dictionary of psychology, 2nd Edition. New York, NY: Penguin.

Salas, E., & Driskell, J.E., & Hughes, S. (1996). The study of stress and human performance. In J. Driskell & E. Salas (Eds.), *Stress and Human Performance*, pp. 1-45. Mahwah, NJ: Erlbaum.

Saavedra, R., Earley, P.C., & Van Dyne, L. (1993). Complex interdependence in task-performing groups. *Journal of Applied Psychology, 78(1),* 61-72.

Searle, B., Bright, J., & Bochner, S. (2001). Helping people to sort it out: The role of social support in the Job Strain Model. *Work and Stress, 15(4),* 328-346.

Serfaty, D., Entin, E.E., & Volpe, C. (1993). Adaption to stress in team decision making and coordination. In *Proceedings of the 37th annual meeting of the Human Factors and Ergonomics Society* (pp. 1228-1232). Santa Monica, CA: Human Factors and Ergonomics Society.

Tepas, D.I., & Price, J.M. (2001). What is stress and what is fatigue? In P. Hancock & P. Desmond (Eds.), *Stress, workload and fatigue,* pp. 607-622. Mahwah, NJ: Erlbaum.

Urban, J.M., Weaver, J.L, Bowers, C.A., & Rhodenizer, L. (1996). Effects of workload and structure on team processes and performance: Implications for complex team decision making. *Human Factors, 38(2),* 300-310.

Weaver, J.L., Bowers, C.A., & Salas, E. (2001). Stress and teams: Performance effects and interventions. In P. Hancock & P. Desmond (Eds.), *Stress, workload and fatigue,* pp. 83-106. Mahwah, NJ: Erlbaum.

Simulation-Based Acquisition

Using Modeling, Simulation and Visualization in the Exploration of Future Navigation Concepts

Joakim Dahlman[1] and Johan Jenvald[2]

[1] Swedish Defence Research Agency
P.O. Box 1165
SE-581 11 Linköping
Sweden
joda@foi.se

[2] Visuell Systemteknik i Linköping AB
Storskiftesgatan 21
SE-583 34 Linköping, Sweden
johan@vsl.se

Abstract

The transportation of people and freight of cargo increase as the numbers of high-speed vessels and huge cargo ships steadily grow. In addition, large cargoes of dangerous bulk freights are shipped across our seas and through the shallow waters of our archipelagoes. This emerging situation calls for new means of navigation concepts in order to support the man on the bridge. In this paper we describe how modeling and simulation can be used to explore possible approaches to future navigation. We emphasize the importance of keeping in mind the realities of life at sea, whenever researchers and developers use modeling, simulation and visualization to present innovative concepts for professional practitioners.

Introduction

The number of high-speed vessels that operate in our archipelago and around the coastline is steadily increasing. The vessels range from military landing crafts to modern ferries that carry hundreds of passengers. There are increased requirements on safety at sea, but at the same time, the naval organizations are trying to cut costs by minimizing the number of crew-members required to operate the vessels and ships at sea. New technologies that can retrieve high-resolution terrain data above and below sea level together with accurate geographical localization systems make it interesting to explore new concepts for navigation at sea. In particular, it is interesting to investigate new concepts that can lower the mental workload for an operator that has to steer a boat safely in shallow waters in the archipelago during harsh weather conditions.

In this paper, we present how we have used modeling, simulation and visualization to explore future navigation concepts together with subject matter experts and users from the Swedish Naval Forces. During our work we have identified a number of crucial aspects of high-speed navigation that affect the workload of the operators and that must be taken into account when defining the requirements of future navigation and support systems. The results from the study include examples of how it is possible to visualize routes, waypoints, dangerous waters and suitable landing spots through a combination of visualization techniques and traditional means of navigation.

The way to success lies in a close relationship between practitioners and researchers during the research and development process. In our research, we have analyzed the current navigation and maneuver situation for the crews of the high-speed landing crafts used in the Swedish Amphibious Corps. We have also used focused interviews and group discussions in combination with sketches and virtual prototypes. To avoid the risk of losing contact with the realities at sea when using modeling and simulation, it is necessary for the researchers to continuously participate in field trials and to collect data from a real environment.

We conclude that our approach has resulted in improved understanding of the strengths and limitations of various future navigation concepts. Used with judgment, modeling, simulation, visualization and synthetic environments can

Figure 1: The interior of a Swedish high-speed boat used by the Amphibious Corps. The navigator sits on the left side and the steersman sits on the right side. Photographer: Joakim Dahlman.

definitely be cost-effective means of bringing new ideas to the practitioners and subject matter experts, in the effort of finding the requirements of tomorrow's navigation systems.

Exploration of Future Concepts

Modeling, simulation and visualization are an alternative way of presenting new theoretical concepts to practitioners. It can be difficult to explain new ideas and concepts through textual requirement descriptions. Various visualization techniques, ranging from simple sketches to the visualization of advanced computer models, can support the work when researchers and system developers want to test new ideas or concepts on practitioners.

Using computer-based tools and virtual environments it is possible to relatively fast try innovative concepts and virtual prototypes in relatively realistic environments. This approach can be cost-effective, because the prototype or concept can be evaluated before it exists in the real world (Jenvald & Morin, 1998). Also the combination of real and simulated entities or simulated artifacts in real operational environments can be of great help in the investigation of the tactical value of new technical systems (Jenvald, 1999; Woods, 1998).

The method used in this study started with the needs and experiences of the end users, and resulted in a future navigational concept, ready to meet future trends and demands. Future navigation, in this case, is likely to change in several aspects. The increasing amount of sensors and new equipment together with a changed tactical appearance will affect the navigation procedure. The driver of a high-speed vessel is supposed to maintain speed, flexibility and navigational accuracy despite being the only person involved in driving the boat. In this case, two operators manage the vessel; one steering the boat

Figure 2: A Swedish high-speed boat operating in shallow waters in the Swedish archipelago. Photographer: Bjarne Svensson.

(i.e. managing helm and speed) and the other is handling the navigation (i.e. giving navigational orders to the helmsman). The future allows us to consider that the navigator will be given complimentary command and control tasks within the framework of the new network based defense.

In order to keep navigation safe, and efficient, the helmsmen must be supported technically. With regards to this given scenario we began our work to identify the basics in handling small high-speed vessels within the Swedish Amphibious Corps. We talked to five operators and interviewed them regarding their skills, experiences and problems. In this way we built up an understanding of the information needed to support the helmsman. We used the information given to us at the first meeting with the domain experts to visualize the first scenario. In this visualization all the navigational information was presented through a helm-mounted display that was only visible to the helmsman. Next, we presented the visualization to the operators and expertise in order to gain their opinion and spontaneous comments. After the second meeting we modified the visualization and we also added some extra features.

With the visualization of today's navigational principles in a new format we began the second phase. We showed the visualization to the helmsman through a helm-mounted display, without any verbal information from the navigator. In this phase, we also wanted to further reduce the amount of information presented to the helmsman, and thereby only present him with the most essential information.

With the future demands from more sensors and additional tasks in mind, we created a second concept based on only the most essential navigation information. We identified this information during the interviews in the first phase. The second phase resulted in a new visualization, which we presented to the same operators and subject matter experts.

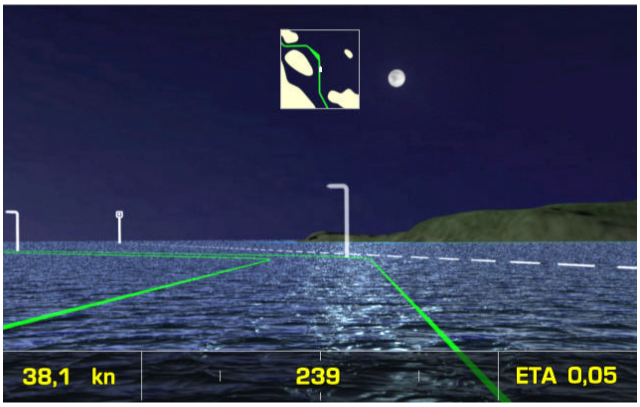


Figure 3: A “lamp post” like arrow illustrates the turning point. The green lines indicate the safe path along the planned route. Presented in the centre are also the navigational information from the electronic chart and the display system (ECDIS). Other vessels are indicated by a vertical line and a box at the top. This indicator can be complemented with the call sign of the vessel or the transponder identification number (AIS).

We received additional comments and considered these comments when we updated the second visualization. The visualization now included less information and a slightly modified interface.

Although the operators at first did not think it would be possible to reduce the amount of information and keep navigational safety and flexibility, they felt comfortable with the new and improved interface. After getting used to the second interface, the operators thought that it would give the same support as the first one.

The two navigational principles will now be implemented and evaluated under both simulated and real conditions.

Developed Navigation Concepts

Traditional navigation follows several principles that are based on information such as when to turn, what to steer at (landmarks), next course and looming dangers. These principles are illustrated in Figure 3 and 4

The next turn is visualized by the “lamp post” arrows and indicates the direction of the turn, and when to turn. By visualizing the turning point with this chosen figure the experts, participating in the study, agreed on the importance of making it apparent even during bad weather conditions. Even if the turning point is visual by only looking at the green path the information might be lost if the weather makes it difficult to identify the path and thereby finding the turning points. By making all the turning points visual we wanted to support the helmsman with enough information so that he could plan the voyage faster since the green path only makes sense at the present leg.

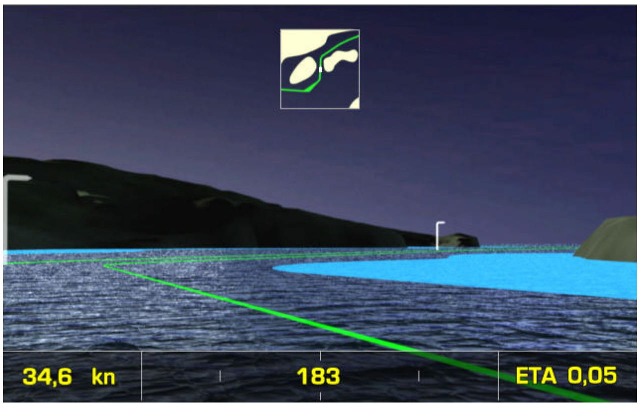


Figure 4: The blue areas indicate shallow waters, less than three meters deep. The navigational information at the bottom of the screen shows from the left speed in knots, heading in degrees and the estimated time to the next turning point in seconds.

The landmark information used in traditional navigation in the archipelago is replaced with the green path in the visualization. The green path indicates the route to follow and therefore the navigator or helmsman does not have to look for other landmarks in the environment, which sometimes can be difficult. Since the navigational cues are different depending on the time of the day, these “guide lines” are supposed to support the helmsman regardless of time. The green path also presents the next course and the course indicator in the middle follows the head movements and is thereby used as a bearing compass.

Possible dangers are marked with blue color, as in the standard sea chart and only for depths less than three meters, due to the draught of the boat, which in this case is 0.6 meters.

At the bottom of the picture the speed in knots, is presented to the left, and at the right is the estimated time to the next turning point in seconds. This information is additional to the original navigational order. Under normal conditions the navigator only tells the helmsman when he is at the turning point, but by visualizing the ETA we enhance the information if the external conditions make the other visualized cues hard to detect.

In the second phase of the visualization, we wanted to reduce navigation information and still maintain safe navigation and flexibility. In this case we made a few examples of a navigation information setup and presented it to the experts as drawings. After having received input and comments we used the same visualization as in the first phase but removed the turning point indicators. We also deleted the green paths, the depth curves, the speed indicator and the ETA.

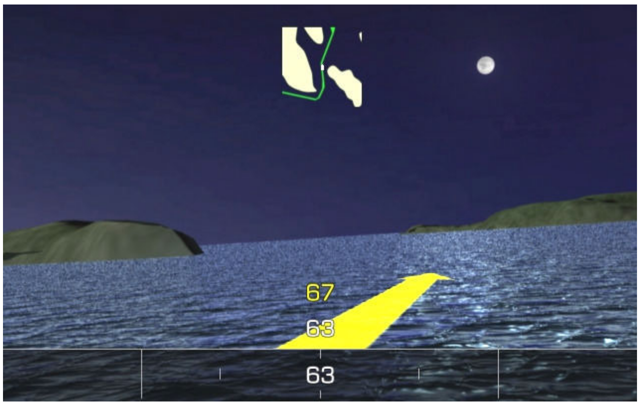


Figure 5: The result of the second phase showing a far more reduced information display than the previous one. In this picture the arrow, indicating the next turn and course is presented (in yellow) together with present course/heading and bearing compass at the bottom

As described previously, much of the information presented in the first visualization is redundant and available by looking at different areas of the screen. For example the green path works both as a course and heading indicator, a danger avoidance tool and a turning point indicator. The preconditions are that the external conditions are good and that the information from the sea chart is reliable. However, the first phase was only intended to visualize today's navigational principle and not to think of any alternative solutions. The second visualization shows a far more reduced information display and only consists of four standardized directional arrows, present course and heading, bearing compass and the next course. The information display is also complemented with the same electronic chart, shown in a top-middle window as in the first visualization.

We decided to use four standardized arrows indicating turns within 40º ± 20º (as illustrated in Figure 5), turns within 90º ± 20º and 90º ± 20º turns. The final arrow indicates straight course and is presented when the actual course and the intended course are the same. In the visualization the arrow is fixed, straight up, as long as you are on course. The arrow starts blinking in the next course direction a few seconds before the next turn. The next course is also presented in yellow at the turning point. When the arrow gets straight, the next course indicator disappears, leaving only the presentation of the intended course and the bearing compass, which is displayed, at the same position as in the previous visualization. As long as the helmsman is focused on keeping the intended course and heading as close to each other as possible, the boat is clear from dangerous waters.

Depending on the distance between the turns and the actual speed, the preparation time is different and the presentation of the blinking arrow can vary from 2-6 seconds.

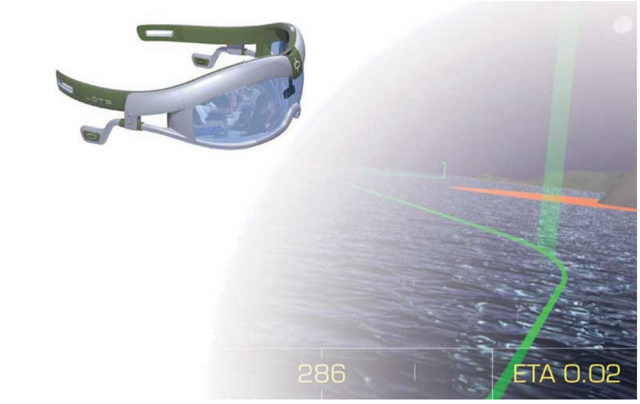


Figure 6: This figure illustrates a possible information-presentation media for the two developed concepts. In this case the information is presented through a pair of glasses. Navigation information is projected inside the glasses, which enables the helmsman to still see the real world through the glasses. The concept supports the use of a bearing compass controlled by head movements.

This gives the helmsman the possibility to reduce speed if navigation is complex. By looking at the electronic chart display he can also plan ahead depending on the complexity of the route.

The change of the representation of the navigational principles to such a large extent as the difference in our two concepts calls for an evaluation. The first step is to implement the two concepts in a virtual environment through the use of a full-scale navigation simulator. By doing this we hope to further develop the two concepts and ultimately move them to the real context and sea trials. Initially the information will be presented on a screen in front of the helmsman at the position for a head up display. At the same time, we will test the second navigational concept together with a head mounted see-through display as shown in Figure 6.

Related Work

In the effort to bridge the gap between cognitive task analysis and system design, Potter and colleagues (2002) introduced the use of intermediate design artifacts. The domain studied was military decision support. In addition, Woods (1998) stated that "From a usefulness perspective, prototypes function as tools for discovery. By changing the tools and representations that people use, we can learn about the demands the domain places on any set of practitioners, the strategies practitioners develop, and how they shape artifacts to function as tools." (p. 171). Woods (1995) also presented related research about support for cognitive work and the aid for human cognition.

In the naval domain Porathe and Sivertun (2002) explored the possibility to use three-dimensional geographical data to support the officer at deck with a "bridge-eye perspective" in the effort to lessen the mental workload for high-speed craft pilots.

Another aspect of using modeling, simulation and visualization is when discussing possibilities and consequences of crew reduction or manning optimization. In this case effects are made visible at the early stages and can there be analyzed in a better way from different perspectives such as mental workload, decision-making, safety etc. (Russell 2000).

Visualizing mental workload is often a result of analyzing tasks performed by an operator and underlying decision-making processes (i.e. collecting information from different sources to create a solid base for decisions) (Werkhoven, Post & Punte 1997).

Discussion

This work has illustrated the benefits of using modeling, simulation and visualization in the early design stages of, in this case, a high speed navigational support system. It has also pointed at the importance of involving the end users in the design process by starting from a "needs" perspective rather then from the technical assets (Dahlman & Jenvald, 2003). The method supports the operator in the strive for optimal requirements to fulfill his or her task in a sufficient manner. One essential point during this kind of work is to make the experts concentrate on the basic navigational needs that underlie their performance instead of focusing too much on the technology used to support their work. By doing this, we manage to reduce the risk of letting the operators be affected by the "state of the art" technology, which is not synonymous with good performance and support.

Another important aspect is to realize that life at sea and navigational performance is far more complex than illustrated in this visualization. Navigation and ship handling is often performed under more difficult conditions and can be affected by many different factors which are out of human control. The work presented in this paper does not include the preparations that form the basis of navigational work. Another aspect is the occasional re-planning, in response to unexpected events that require other skills than those illustrated in this example. All missions performed today are being preplanned and prepared before leaving port, if information regarding final destination and waypoints along the way is available. If re-planning is necessary the navigator usually does this under way and compensates by reducing speed or stopping, if possible. In order for the helmsmen to perform re-planning under way he must be equipped with decision support systems that give alternative routes based on experience and intelligence. Navigation in this case is performed under two different modes, depending on time and weather conditions and the first principle is optical navigation, which is the primary method. The secondary method is radar navigation and this method is used under conditions with poor visibility.

During our work we have received important lessons learned through the reactions and reflections on our computer visualizations from the navigators of today's high-speed boats. It is important to remember that further studies also have to address the question how new navigation concepts and possible crew reductions affect the overall mission planning process. Even if some of the explored navigation concepts seem promising, it is important to broaden the perspective and analyze the overall performance of the mission. In this case, high-speed navigation is one of several important factors that has to be considered in order to improve the understanding of the situation for the crew of a future high-speed vessel.

Future work includes taking the developed navigation scenario to an interactive virtual environment, to let the operator act in the synthetic world. Future studies will thereby further investigate whether the reduced information visualization is sufficient for keeping navigation safe and flexible. By using simulators for navigation training we can further develop the visualized example before taking it to the real environment and sea trials.

Conclusions

We argue that modeling, simulation and visualization have great potential when it comes to the illustration of new theoretical concepts and in the early phases of research and development. However, it is important to understand the limitations of modeling, simulation, visualization and virtual environments and use them with care. We conclude that it is necessary to have great knowledge in the studied domain and that field trials in real environments are invaluable and necessary in order to keep up with the requirements of reality.

The way towards success lies in a close relationship with real practitioners during the research and development process. Used in the proper way, modeling, simulation and visualization can decrease the gap between users, researchers and system developers and support more cost-effective development of future systems.

References

Dahlman, J. & Jenvald, J. (2003). Operatörer i marina system- stridsledning med fusionerad sensordata [Operators in Naval Command and Control System using data fusion and sensor data] (in Swedish). FMV Proj. TU 23321:11122/03

Jenvald, J. (1999). Modelling and Simulation – an Instrument for the Development of Tactics and Technology. *The Journal of the Royal Swedish Society of Naval Sciences*, 162(3), 291-297.

Jenvald, J. & Morin, M. (1998). Tactical Evaluation of New Military Systems using Distributed Modelling and Simulation. In *Proceedings of the 12th European Simulation Multiconference (ESM'98)*, pp. 279-284, June 16-19, Manchester, UK.

Porathe, T. & Sivertun, Å. (2002). Information Design for a 3D Nautical Navigational Visualization System. In Proceedings of The Eighth International Conference on Distributed Multimedia Systems, September 26 – 28, San Francisco, California.

Potter, S., Elm, W., Roth, E., Gualtieri, J., & Easter, J. (2002). Using intermediate design artifacts to bridge the gap between cognitive analysis and cognitive engineering. In McNeese, M., & Vidulich, M. (Eds.), *Cognitive Systems Engineering in Military Aviation Environments: Avoiding Cogminutia Fragmentosa!: A report produced under the auspices of The Technical Cooperation Programme Technical Panel HUM TP-7 Human Factors in Aircraft Environments (HSIAC-SOAR-2002-01)*. Wright Patterson Air Force Base, OH: Human Systems Information Analysis Center.

Russell, K.J., Rethinking reduced manning design and optimization using a modified system approach, United States Coast Guard, March 2000.

Werkhoven, P.J., Post, W.M., & Punte, P.A.J. (1997). Validation of ADCF bridge concepts using virtual environment techniques. Ship control systems 11:th symposium. Southampton, Vol 2.

Woods, D. D. (1995). Toward a theoretical base for representation design in the computer medium: Ecological perception and aiding human cognition. J. M. Flach, P. A. Hancock, J. Caird, & K. J. Vicente (eds.). *Global perspectives on the ecology of human-machine systems, Vol. 1. Resources for ecological psychology*, pp. 157–188, Lawrence Erlbaum.

Woods, D. D. (1998). Designs are hypotheses about how artifacts shape cognition and collaboration. In *Ergonomics* 41(2), pp. 168-173.

SBA Applied: Bridging Gaps in the Development Process

Örjan Blom

ISD Datasystem AB
Box 135, SE-182 12 Danderyd
oerjan.blom@isd.se

Abstract

The vision of Simulation Based Acquisition (SBA) is an acquisition process in which Defense and Industry are enabled by collaborative use of simulation technology that is integrated across acquisition phases and programs. SBA is very convincing, presented as a theoretical framework, but an optimal path to applying the concept has yet to be formulated. Little has been stated so far on the possible risks, difficulties and side effects of trying to adapt existing development organizations in order to start using SBA. The SBA concept is moving forward, but in order to create the evolved culture, refined systems acquisition process and advanced systems engineering environment envisioned, progress must be coordinated and experiences from applying and using the framework must be shared within the research and development community. This paper illustrates a potential problem that can present itself as theory is put into practice. During acquisition, requirements are modeled to constitute an interpretation of business goals, business rules, users' needs, target environment etc. Further interpretations of this model are then made throughout the acquisition process during technology and product development. Making interpretations of an interpretation in this way can be troublesome, since there is often a gap between how the model is perceived at different stages of development. The gap resides in that people involved at the different stages often do not share the same tactical or technical knowledge. Surrounding factors are perceived and/or valued differently. There are also other factors that can combine to increase the distance. Different companies might be represented, with different goals and (perhaps even conflicting) strategies, which will affect interpretation, realization, testing and validation of requirements. Facing this risk, the expressed need for a centralized coordination of methods and efforts is stressed, as well as an adapted development process that can be integrated into SBA. The process must allow results and experiences to be shared across the development cycle to optimize work and create common ground for people working at each stage. One way to achieve this is through posing specific requirements on the documentation that is produced and distributed, viewing it as a tool for communication. Another is to encourage certain "violations" to formal procedures, allowing shortcuts and face-to-face interaction between system analysts and system developers. The goal is to bridge the gaps in the acquisition process by helping people to direct their work towards each other. Sharing experience in this way, and across the development community, will make the goals and expectations on SBA more likely to be reached.

Introduction

This section gives a brief introduction and overview of the concept Simulation Based Acquisition – the vision, its implications and prerequisites.

The vision of Simulation Based Acquisition (SBA)

SBA is an acquisition process in which Defense and Industry are enabled by robust, collaborative use of simulation technology that is integrated across acquisition phases and programs (ARO, 2000; Hollenbach, 2000; SBAISG, 1998).

The United States Department of Defense (DoD) began developing this concept in order to draw upon the benefits displayed by Industry in using modeling and simulation for systems acquisition.

The goals were, and still are, to reduce the time, resources and risk associated with the entire acquisition process and increase the quality, military worth, and supportability of fielded systems while reducing total ownership costs throughout the total life cycle, and enable Integrated Product and Process Development (IPPD) across the entire acquisition life cycle.

The key features of SBA, in comparison to traditional acquisition processes, are:

- Modeling of requirements with the help of end-users, and the use of models as a communications medium between end-users, developers and other stakeholders.
- Simulation of the system's operating environment, including complex system of systems (SoS) interactions.
- Using modeling and simulation (M&S) integrated with test, verification and validation (V&V).
- Interchanging information among heterogeneous tools and interoperating these tools via open standards.

The use of M&S, early and throughout the acquisition life cycle, gives an increased confidence that relevant requirements are captured and fulfilled, through end-user involvement and traceability of requirements.

Integrated with V&V, modeling and simulation is used to facilitate learning, assess technology maturity and interoperability, facilitate integration into fielded forces, and confirm performance against documented capability needs and adversary capabilities as described in the system threat assessment[1]. M&S also allows for risk management, in that risks are exposed and considered earlier in the process (Price et al., 2002).

Interoperability supports reuse and enables different development organizations to work on the same product in parallel.

Prerequisites for implementing SBA

Based on the US DoD initiative, many studies, documents, workshops and presentations have sought to establish a road map for implementing SBA (Acquisition Council, 1998; USD(AT&L), 2002). This roadmap has yet to reach its final form, but factors, on which the success of SBA depend, have been identified. These are some of the most important (ARO, 2000; Hollenbach, 2001):

- Legislation, policy, budgeting and management changes, in order to help share information and tools and gain interoperability between development organizations and create a collaborative environment between Defense and Industry.
- Accurate and comprehensive assessments of design, manufacturing, employment and support concepts earlier in the acquisition cycle, through the iterative use of modeling and simulation, in order to reduce time, resources and risk, and allowing a system to be designed, built, tested and operated in the computer before critical decisions are fixed and manufacturing begins.
- Collaborative environments, in which off-the-shelf (or minimally modified) sets of reusable, interoperable tools and supporting resources are used to assess an emergent technology, capability or concept.
- Verification and validation in the broader context of an expected real-world environment, with documentation to facilitate accreditation and reuse.

SBA for the Swedish Armed Forces

The Swedish Armed Forces are now beginning to show an interest in SBA. Reducing time, resources and risk sounds promising in a situation where massive budget cuts are being made while the organization is in great need of acquiring new systems – organizational and technological – to meet the requirements of a changing real-world situation.

The Swedish Armed Forces are transitioning from the traditional invasion deterrence defense to a mission-oriented Network Centric Warfare (NCW), but the current NCW-vision requires considerable clarification and needs to be defined in greater detail in order to be realized. The process has been said to be very technology-driven, with a lack of understanding for how technology relates to human action (Persson, 2003). Dependence on external competence, with a reduced need for intra-organizational knowledge, is a suggested reason for being in this situation. However, relying on Industry to provide technical solutions alone does not yield technology-driven systems acquisition. If objectives are abstract, as in the NCW-vision, and the Defense leaves it to commercial companies to interpret, define and develop the vision, the solutions will not necessarily assume the needs of the Defense. Instead, each company will try to create a need that matches their own unique design, development and production capabilities. Competition is thus not primarily directed towards providing the customer with a solution optimized for the customer's needs, but towards outmatching other companies in conceiving and developing the most advanced technology for the lowest price, which will create a need for the customer. Because systems acquisition has largely depended on external competence, this competence is still needed in developing the NCW. The vision, concepts and solutions should therefore be the result of collaboration between Defense and Industry. The Swedish Armed forces, however, has to

[1] In accordance to the Department of Defense directives 5000.1 and 5000.2, issued by the USD(AT&L) (2003a; 2003b).

insist on a methodology that will guarantee that end-user and stakeholder needs are prioritized.

SBA could supply the remedy to these problems, since the process requires that an accurate model of stakeholder and user needs is established before assessments of design and manufacturing can begin to be made. This ensures that stakeholder and end-user needs are put in the first room. Visions have to be defined in detail and concretized before development can begin. By using modeling and simulation early and throughout the acquisition cycle, needs and requirements are also continuously updated, verified and validated.

Assuming SBA as the framework for systems acquisition, competition can be directed, not towards inventing the most advanced military technology, but towards modeling and tailoring systems and solutions for the Swedish Armed Forces.

Establishing a Swedish road map

SBA should be a joint effort between the Swedish Armed Forces and Industry. This is necessary if the prerequisites for implementing SBA are to be fulfilled. The ability to interchange and reuse competence, experience, technology and tools is fundamental when attempting to reduce the time, resources and risks involved in the acquisition process.

How to practically enable this collaborative environment is not equally easy to formulate. The talks and discussions concerning NCW are on a very high level of abstraction and consist to a great extent only of visions and ideas.

There are many theories and a lot of experience that come from other defense organizations, but how to establish an SBA road map for specific Swedish needs is far from obvious. Parallels to other efforts can probably be drawn but it is not safe just to copy other SBA-initiatives, since the process must be adapted to the unique domain and real-world situation. Decisions on how to design the process, what methods should be used, how the implementation of SBA should be organized and how Defense and Industry should collaborate must be made. So far, there has been little exchange of information regarding these issues, which means that a lot of work remains before the prerequisites of a functional SBA can be reached, at least at a Defense organizational level.

Considering this, a pure theoretical approach will probably not be enough, and all development efforts cannot simply be put on hold, waiting for a solution to transpire. It is difficult to go directly from theory to practice and foresee all outcomes. Initiatives must therefore also take place on the more practical level of systems engineering.

Practice makes perfect, and by venturing projects using an SBA-approach, the process can be modeled, simulated and evaluated in the broader context of the real-world environment. Using "SBA" on SBA will help to gain valuable experience in the field, and give an opportunity to optimize the process to fit with current demands and situation, and make changes along the way.

The projects should serve as test-platforms on which new ideas, concepts, standards and methodology can be formed, elaborated, tested and shared.

SBA at ISD

How then, should SBA be practically implemented in a development project? Most development organizations already have elaborated ways of designing and constructing systems, and many of them are already using M&S in their work. This makes it difficult to generalize and give a general recommendation. Instead, a specific example is given here, so that readers may draw their own conclusions from their own perspectives.

ISD Datasystem AB is an independent commercial systems development company that works within the Defense Industry. ISDs SBA projects have been directed primarily towards technology development in the form of prototyping. Technology development, separated from product development, has the objective of reducing the risk regarding integration and manufacturing by bringing a technology up to the point where it can be readily integrated into a new product and counted on to meet requirements (GAO, 1999). The domain of interest has been the design and prototyping of future interactive command and control systems for the Swedish Air Force (FMV, 2001; FMV, 2002; FMV, 2003), and Network Centric Warfare.

Presented here is ISDs view on SBA, and part of the company's experience, from trying to implement the framework into technology development projects.

Complementing SBA with a systems engineering process

SBA describes how systems should be acquired. It is also necessary to go one level deeper and describe how systems should be engineered and developed in order to implement the concept into a development organization. SBA will need to be complemented by a systems engineering process in order to function as a framework for developing systems.

Since SBA is a new concept, even though modeling and simulation have been used extensively in Sweden, there is little practical experience from trying to implement SBA or complete it with a systems engineering process. There is only little information on the possible risks, difficulties and side effects of using SBA. It would be useful to know what problems to be aware of and what to avoid[2].

[2] Price et al. (2002) present an interesting review on what can, and cannot, be expected from SBA.

ISD has constructed a systems engineering process to fit an SBA approach for technology development projects. The process originates from the Rational Unified Process (RUP) (Kruchten, 2000). RUP is a relatively general description of systems engineering, which is advantageous when it comes to adapting it to SBA. Its iterative structure is also well suited for SBA projects.

The ISD process can be described as the iterative application of analysis, design, implementation, test and validation to requirements. Requirements are gathered to form a model, which is realized in a functional prototype. This process is illustrated in Figure 1.

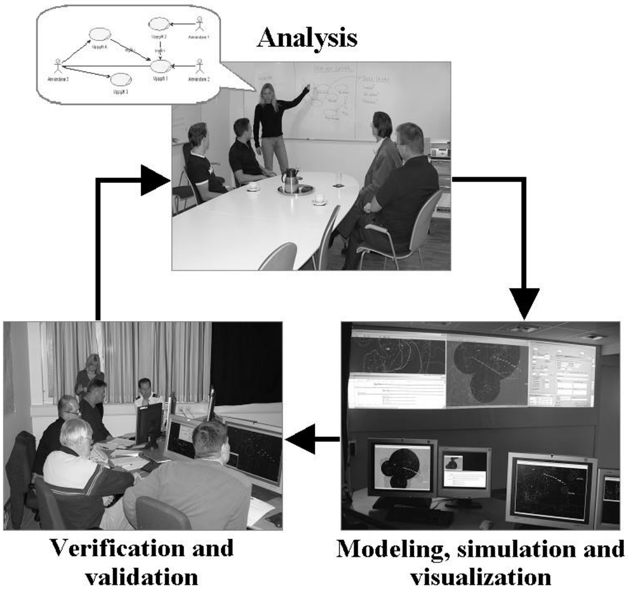


Figure 1: The system engineering process at ISD.

The technological development proceeds through experimentation and evaluation. Requirements and different design solutions are realized in the prototype and then evaluated. Iterative development enables good solutions to be kept and elaborated, while bad solutions are dismissed.

End-users, or subject matter experts, are a vital part of this process, since they serve as an important source of information and a window towards the expected real-world situation.

The end-users help to define the model and prioritize requirements. Later in the process, they are asked to verify if the requirements have been interpreted and realized adequately, and validate if the requirements are adequate for a future system in the broader context of the expected real-world situation. Results are transformed into a requirements specification for the future system.

Using prototyping during the technological development is a relatively cheap way to investigate how to prioritize requirements, design for usability, and optimize costs during the later product development, where the major investments are made (GAO, 1999).

Sharing experience from implementing SBA into a development project

As stated earlier, there is, thus far, little practical experience shared within the community when it comes to implementing SBA. Experience must be discussed, in order to find a rigorous, empirically proved process and methodology that is guaranteed to get the job done. This particularly concerns negative experience and appropriate solutions.

This section describes one problem that most certainly will be encountered in one way or another during systems engineering and the implementation of SBA. Furthermore, a solution, or a set of solutions is presented, which has helped ISD in the process of engineering systems[3].

Gaps in the development process

During acquisition, requirements are modeled to constitute an interpretation of business goals, business rules, users' needs, target environment etc. Further interpretations of this model are then made throughout the acquisition process during technology and product development. Making interpretations of an interpretation in this way can be troublesome, since there is often a gap between how the model is perceived at different stages of development. The gap resides in that people involved at the different stages often do not share the same tactical or technical knowledge. Surrounding factors are perceived and/or valued differently.

For example: it is often not easy for a developer to understand the reason for why requirements are formulated in certain ways and envision the final result, when the iterative process means that requirements are given one at a time. The consequence of this is that when certain requirements cannot be implemented in the exact way specified, the developer's alternative solution and implementation may not comply with the big picture. It is also not possible to check up on every function or detail that cannot be readily implemented. This should be part of the developers "artistic freedom".

Another issue, which is related to the previous, is that the developers might not get enough constructive feedback on their performance from what the end-users report during validations. The developers just see new versions of the

[3] The SBA projects that ISD has endeavored have so far been on a somewhat limited scale. The featured problem is probably greater in a large project, and it remains to be tested whether the proposed solutions can be scaled to fit a larger project.

requirements, without knowing why the requirements have been changed. This means that there will be difficulties in building an interpretation of the model and adjusting it to be correct, deciding where to allocate the main efforts, and understanding what will be of use and give effect for the end-users, since this is not always related to the technological complexity of the system.

Gaps can also arise as a consequence of separating technology development from product development. This is actually an important aspect of SBA, in that technology can be evaluated through the use of M&S before fixing on a product solution[4]. However, making this distinction may create a gap between the two stages of the acquisition process. Different companies might be represented, with different goals and (perhaps even conflicting) strategies, which will affect interpretation, realization, testing and validation of requirements. For instance, a product development organization often has an ambition to reuse components and code, which might color how requirements are prioritized and implemented. There are also often differences regarding the people that are involved, their technical and tactical knowledge, hardware, tools and standards.

Bridging the gaps

Facing these issues, it is necessary to come up with appropriate solutions and adapt the systems engineering process accordingly. The following sections will elaborate on how ISD has succeeded in bridging gaps in the development process.

Evolved requirements management. During modeling, the functional and non-functional requirements that are gathered are prioritized, broken down and translated into system requirements. The system requirements are organized into components, in order to form a component based architecture.

A component is a self-contained, reusable building block that can be used independently or assembled with other components to satisfy requirements. A component handles a specific event, or related set of events, and provides a particular function or group of related functions through an interface. The component interface may send or receive data from a file, or may be a user interface.

One increment corresponds to an iteration of a component, with analysis, design, implementation and test. The components can be iterated independently of one another. Increments can also be analyzed, designed and implemented in parallel. It is very important, however, that there is only one version of every component for each increment, in order to minimize the risk of confusions.

The result of the iterative and incremental development is a process that is more flexible than traditional processes in changing the representation of requirements. Increments, being less extensive than a full iteration, tighten the process by reducing cycle-times, and distance between stages of development and the people involved.

One good example of this is that sometimes, and sometimes quite often – especially during technology development – the requirement administrators will realize that a component will have to change, compared to the version that has already been forwarded to implementation. It is then of great help if the requirement administrators are fully aware of how far into the process of implementation the component is. If they know that none or only little implementation has been made, a "violation" can be made to procedures so that the current component increment is withdrawn and skipped in favor of the next.

Often, several increments to a component are planned in parallel. It is then useful for the developer if early increments state something about what will happen later, even if each increment is quite small. This will reduce the risk for ending up in a situation where the developer may say: "Had I known that from the beginning, I would have done completely differently, but now it is difficult to change anything". This also applies to requirements that are uncertain. If there is a risk that requirements will change later, this should be stated early.

Procedures like this are normally not described in systems engineering or systems development processes, but are absolutely necessary and serve as valuable methods in order not to lose time and money. The risk of developers losing their motivation, because they experience frustration in throwing away code that has required a lot of effort and commitment to build, is also reduced.

As stated earlier, this is particularly important when it comes to technology development, when sudden and major changes to requirements are not uncommon.

Configuration management, and documentation as a communications tool. The procedures mentioned in the previous section pose specific requirements on the documentation that is produced during the acquisition process.

There is a necessary formalism in a project when it comes to configuration management and documentation, which increase with system complexity and number of project members in order to maintain control. This can form a gap between project members if they do not get the opportunity to meet face to face. The problem occurs because it is very hard to formulate unambiguous requirements and explain the big picture, context, relevance and essence of requirements in writing. A reason for this is that the requirements are based on a model that contains only what was considered important to extract from the real-world situation, from a system requirements perspective.

[4] The distinction between technology and product development is recommended by the United States General Accounting Office (GAO) (GAO, 1999).

Due to iterative and incremental systems engineering, the need for formal documentation decreases, in favor of an increased need for configuration management, that is, a central coordination of requirements and components. Requirement administrators have to be updated on the progress and status of all the components, in order to enable increments to be withdrawn, updated, and worked on in parallel. Thus, the formal documentation is reduced while interaction and communication between project members is encouraged, using the documentation of individual components as a mediating tool.

However, it is important that the documentation supplies developers with relevant context. By providing information about the purpose of the component and how it will function once deployed, from the end-user's perspective, the developers can form an idea of the best design for the product. It is then easier to go in the right direction and create correct system architecture. A document standard that permits and encourages this information is important.

Configuration management requires the most resources at the beginning of a project. It is therefore less time-consuming than constantly producing formal documents, which are seldom written voluntarily and that only give little outcome compared to the effort. Project members usually favor this solution.

Sharing information and providing feedback. What does it mean to explain the big picture, context, relevance and essence of requirements?

It means that it will be easier for project members to keep track of each other, and see how their own contributions fit in with the big picture, and have expectations on what the next logical step in the development of each component will be.

By using iterative and incremental development, the project members will have to communicate more often, which yields an opportunity to answer and put forward questions, explain circumstances concerning requirements and describe the context in which decisions are made. Results and experience must be shared between project members in order to provide feedback that can be used to optimize work and help create common ground. This will increase the total competence of tactical and technical information and help everyone involved to direct their work towards each other and alleviate each other in a natural way, helping them to function better as a team.

Developers should also be given an opportunity to meet the end-users, and create an image of the people that will use the system in the end. Documentation will never be able to communicate this image.

Visualizing requirements. Bridging the gap between technology and product development is more difficult. Communication is generally not very frequent between people working at the two different stages, and therefore the interpretation of requirements that the product development team makes is further abstracted from the end-users who provided the model. At the same time, documentation is not enough in order to provide the essence of requirements.

One way to solve this problem is for the technological development team to build a functional prototype (an illustrator) to display the requirements. Not only does such an illustrator give an idea of the appearance and behavior of the system in a real setting, but it also shows how the documented requirements are integrated to form a system. This gives a completely different perspective when it comes to valuating purpose and practical use of certain solutions and components, and the context in which the requirements were specified.

It may also be helpful if the product development team have a chance to visit the technological development team, to share experience and tactical decisions behind functions and solutions, and meet the people behind the prototype. Such cooperation adds continuity to the acquisition process.

Conclusion

Bridging the gaps in the development process by helping people to direct their work towards each other is important when it comes to building a standardized, efficient and goal-directed systems engineering and acquisition process.

Sharing experience, within each project and across the development community, will hopefully enable a discussion and a debate that is necessary in order to produce a standardized methodology and road map for building an SBA-process for the Swedish Armed Forces and Industry. With this debate the goals envisioned with SBA will have greater expectations of being reached.

References

Acquisition Council. (1998). *A Roadmap for Simulation Based Acquisition – Report of the Joint Simulation Based Acquisition Task Force* (Draft). Acquisition Council, December 4.

ARO (Department of the Navy Acquisition Reform Office). (2000). *Simulation Based Acquisition (SBA) Status and International Implications*. ARO, Washington, D.C.

FMV, (2001). Enquiry (219951 – LB566668). Stockholm: FMV.

FMV, (2002). Enquiry (219951 – LB606666). Stockholm: FMV.

FMV, (2003). Enquiry (219951 – LB623839). Stockholm: FMV.

GAO (United States General Accounting Office). (1999). *Defense Acquisition: Best Commercial Practices Can*

Improve Program Outcomes (GAO/T-NSIAD-99-116). GAO , Washington, D.C.

Hollenbach, J.W. (2000). Department of the Navy (DON) Corporate Approach to Simulation Based Acquisition. *In proceedings of the Fall 2000 Simulation Interoperability Workshop*. Orlando, FL.

Hollenbach, J.W. (2001). Collaborative Achievement of Advanced Acquisition Environments. *In proceedings of the Spring 2001 Simulation Interoperability Workshop*. Orlando, FL.

Kruchten, P. (2000). *The Rational Unified Process, 2nd Edition: An introduction.* Reading: Addison Wesley Longman, Inc.

Persson, P. (2003). On Theory and Methods for the Development of the New Defence. *In proceedings of the 2003 Civil and Military Preparedness (CIMI) Seminars.* Enköping.

Price, S., Coombs, M., Miller, R. (2002). The Reality of SBA – How Simulation Can and Cannot Inform the Acquisition Process. *In proceedings of the Fall 2002 Simulation Interoperability Workshop*. Orlando, FL.

SBAISG (Simulation Based Acquisition Industry Steering Group). (1998). *Simulation Based Acquisition Functional Description Document* (Draft). NDIA SBA Industry Steering Group, September 23.

USD(AT&L) (Office of the Under Secretary of Defense for Acquisition, Technology and Logistics). (2002). Interim Defense Acquisition Guidebook October 30. Washington, D.C.

USD(AT&L) (Office of the Under Secretary of Defense for Acquisition, Technology and Logistics). (2003a). Department of Defense Directive 5000.1 May 12. Washington, D.C.

USD(AT&L) (Office of the Under Secretary of Defense for Acquisition, Technology and Logistics). (2003b). Department of Defense Directive 5000.2 May 12. Washington, D.C.

Towards rapid 3D modelling of urban areas

Ulf Söderman, Simon Ahlberg, Åsa Persson, Magnus Elmqvist

FOI (Swedish Defence Research Agency)
P.O. Box 1165, SE-581 11, Linkoping, Sweden
{ulfso, simahl, asaper, magel}@foi.se

Abstract

Modelling and simulation applications for operations in urban areas often set high resolution and fidelity requirements for the 3D environment models. For applications supporting ongoing operations, e.g decision support, mission planning, etc., additional requirements such as time frame and accessibility to the area of interest must also be taken into account in the modelling task. One possible approach to handle all these requirements is to use methods that allow for automatic 3D modelling using recent data from reconnaissance and surveillance sensors. In future urban operations we can expect to see different types of UAVs and UGVs equipped with sensors like 3D laser radars, IR- and visual cameras. The data from these sensors can be used for environment modelling.

This paper will present new developments in data processing aimed for rapid 3D environment modelling and visualization for e.g. mission planning and mission rehearsal, mainly in urban environments. The methods include ground estimation, building reconstruction and tree identification. The long term objective is an automatic production chain from environment data collection to environment models ready for use in many different applications.

Introduction

Modelling and simulation applications for operations in urban areas often require very detailed and high-fidelity models of the real environment. However, the construction of such models is not straightforward. Common obstacles are the availability of data describing the area and the vast amount of details to be handled during modelling. For applications supporting ongoing operations, e.g. crisis management, decision support, mission planning, etc., additional requirements such as time frame and accessibility to the area of interest must also be taken into account. Thus, some kind of automatic modelling tool is needed (McKeown, 1996).

The long term objective of our work is the development of automatic methods for rapid construction of high-fidelity natural environment models from sensor data (Ahlberg & Söderman 2002). In this paper we will present some recent developments in data processing, especially methods for 3D reconstruction of buildings.

To obtain high resolution data we have used recent airborne laser scanners (ALS) and digital cameras. A short introduction to ALS systems and the type of data they produce is given in the next section. For the construction of high-fidelity models the raw data from the ALS and camera system is processed in order to produce data sets like digital elevations models, orthophoto mosaics, 3D object models and feature data in term of points, vectors, and polygons. For this purpose we have developed several new methods including a method for bare earth extraction based on active contours (Elmqvist, 2002), a method for identification of individual trees, classification of tree species, and estimation of tree position, height and canopy size (Persson et al, 2002, Holmgren & Persson, 2003), and a novel method for 3D reconstruction of buildings. The tree identification and estimation method has been validated using ground truth data (Persson et al, 2002, Holmgren & Persson, 2003). One example of the kind of models that can be produced using data from ALS systems and cameras is illustrated to the left in Figure 1. The right image shows a Digital Surface Model (DSM) based on raw gridded laser data having 0.25 m post spacing overlaid with high resolution aerial imagery.

In the next section we will briefly discuss pre-processing of ALS data, e.g. gridding and classification. We will also briefly illustrate the ground estimation and tree identification methods which already have been discussed in length elsewhere.

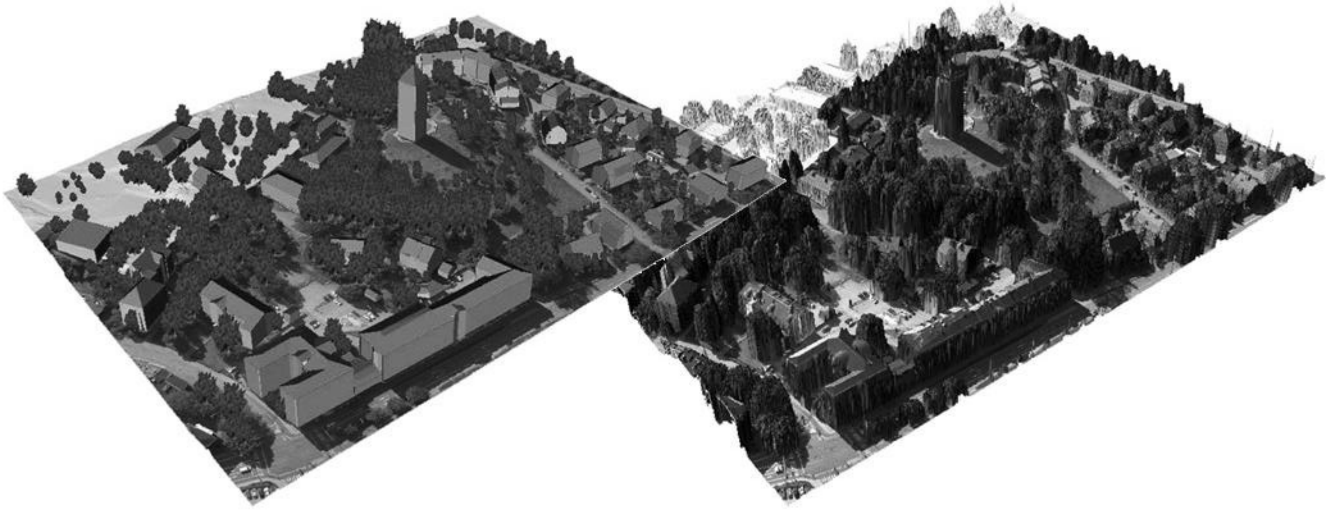


Figure 1. A 3D environment model for e.g. real-time visual simulation based on ALS and aerial imagery (left) and a DSM obtained from ALS data partially overlaid with an orthophoto mosaic (right).

In the third section we will present and illustrate the new method for automatic 3D reconstruction of buildings. In the final section we will illustrate the use of these methods for construction of high-fidelity models of urban areas for real-time visualisation applications.

Lidar and image data

Modern ALS systems consist of a range measuring laser radar (sometimes also referred to as LIDAR - *L*ight *D*etection and *R*anging), a positioning and orientation subsystem using a differential GPS and INS (*I*nertial *N*avigation *S*ystem) and a control unit. The laser radar consists of a laser range finder measuring range using time-of-flight measurement, an opto-mechanical scanner and a control and data storage unit.

The most frequently used scanners are line scanners which deflect the laser beam back and forth across the flight track using oscillating mirrors. The measured elevation points form a zig-zag pattern on the terrain surface as the platform moves forward, see Figure 2. The positions of the points correspond to the locations where the laser pulses have been reflected on the terrain or object surfaces, e.g. on trees, bushes, building roofs, etc. The positions are determined using the measured range and the position and orientation of the laser radar system. The resulting (x, y, z) coordinates are normally given using some suitable reference system.

In addition to terrain elevation, modern ALS systems often measure amount or "intensity" of the reflected laser pulse. This information can be used to produce a monochromatic "intensity image" of the survey area, see Figure 3. Many modern systems also have the ability to detect and register multiple returns for one laser pulse. Usually, two returns are registered (first and last return). Multiple returns occur when different parts of a laser pulse "footprint", which may be several dm in diameter, hit different objects or different parts of one object at different elevations.

Figure 2. Elevation points from airborne laser scanning. Top: data from a forest area (x and y scales are non-uniform for illustration purposes). Bottom: data from an urban area.

For example, one part of a footprint may hit the top of a lamp post and the remaining part may hit the ground below. The intensity and the multiple return data provide additional information which together with the elevation data form an excellent data set for feature extraction and 3D reconstruction tasks.

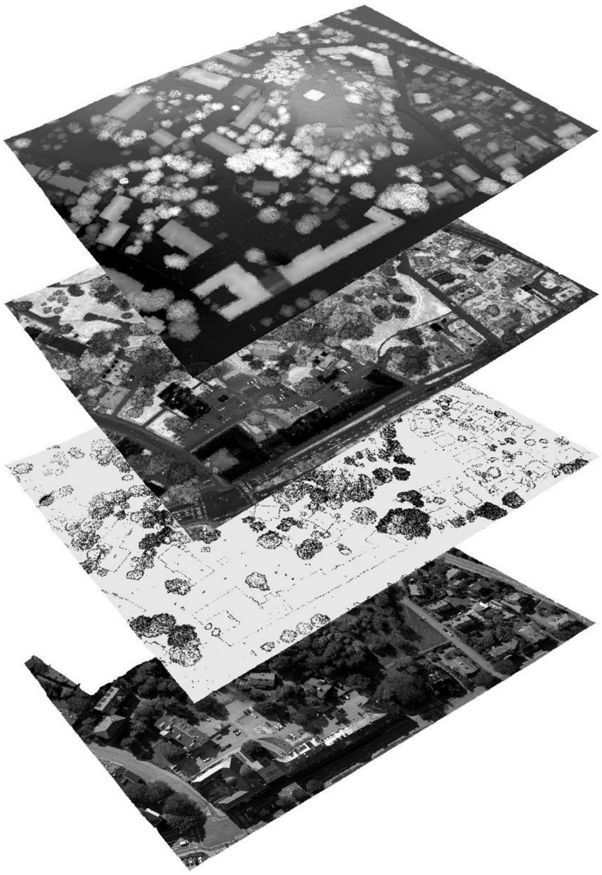


Figure 3. Example of ALS data from a small urban environment (250 m x 350 m). From the top: elevation data, intensity data, multiple return data and an orthophoto mosaic.

In many applications, aerial imagery is a desired complement to the laser data. This can be achieved by extending the ALS system with a high resolution digital camera. Camera position and orientation can be obtained for each image using the GPS- and INS-data from the ALS system. Georeferenced orthophoto mosaics can then be produced using the images, the camera orientation data and a digital terrain elevation model from the laser data, so no stereo overlapping images and ground control points are needed. Therefore, orthorectification can be completely automatic in the future.

In our work we have used laser and image data from the TOPEYE system. This system is operated by the services provider Topeye AB (www.topeye.com). The data used in the work presented here are from different data sets collected at different occasions between 1997 and 2001. The surveyed areas consist of different types, e.g. forests, urban terrain and rural terrain and the point density varies between the data sets from approximately 4 up to 16 points per m^2.

Data processing

The construction of high fidelity environment models requires a number of different input data sets, for example digital elevations models, orthophoto mosaics, 3D object models and feature data in terms of points, vectors, and polygons. A first step to obtain these data sets is to resample the ALS data in regular grids (matrices) and then classify the data in three main classes; ground, vegetation and buildings (Hug 1997, Maas 1999, Söderman & Ahlberg 2002). The data in the ground class may also be further classified as e.g. paved areas (roads, parking lots, etc.) and non-paved areas.

Classification

Since the raw ALS data we typically use has very high point density the data is usually resampled in regular grids or matrices having only 0.25m post spacing. A number of different matrices containing elevation or intensity data are often constructed, e.g. one containing the lowest elevation or z-value in each cell (DSM_{zmin}), and one containing the highest z-value (DSM_{zmax}). In case a continuous “surface” is needed, empty cells are filled using interpolation, see Figure 4.

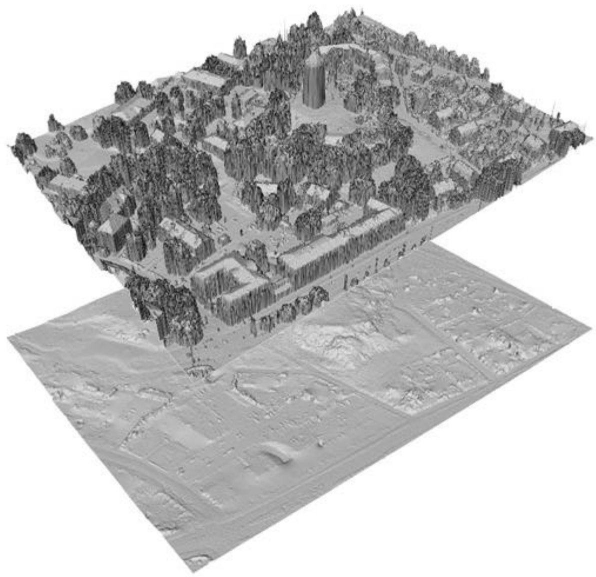


Figure 4. Top: A DSM from a small urban area. Bottom: The resulting DTM constructed using the active contour method. The post spacing in the elevation grids of the DSM and DTM is 0.25m.

Classification of ground points is performed by first estimating a ground surface representing the bare earth

(Elmqvist 2002), see Figure 4. All pixels on or close to this surface are then classified as ground, see Figure 5. To classify buildings, objects (i.e. groups of connected pixels) are first separated. Since multiple returns mainly occur in vegetation and at edges of buildings but in general not within the compact surfaces of roofs, vegetation can be filtered out using multiple returns (i.e. removing pixels containing multiple returns). All groups of remaining pixels with an elevation of more than two meters above the ground surface are treated as objects.

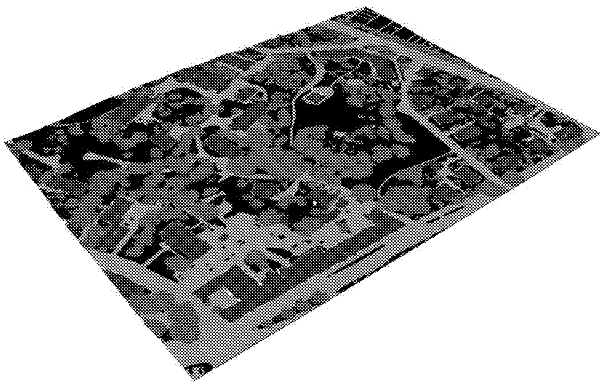


Figure 5. Result of classification, Grey levels (light to dark) indicate as follows: paved ground, vegetation, buildings, and non-paved ground.

Each object is separately classified based on measures of its shape and local variation in height. Man-made objects such as buildings usually consist of continuous, compact surfaces that are bounded by long straight lines. On the other hand, natural objects such as vegetation have irregular shapes without any linear structure and large vertical variations throughout the objects since the laser pulse often penetrates the canopy. The shapes of objects are measured by comparing the histograms of the Hough transform of the objects' contours. Two other measures used are the curvature, i.e. the second order derivative, and the maximum slope for each pixel and its eight neighbouring pixels. The height difference within vegetation varies between neighbouring pixels, and hence the curvature and slope are large in contrast to within flat or tiled roofs of buildings. Large values that occur at edges of buildings are removed in the object separation when using the multiple returns. Based on the three measures, each object is classified as building or non-building (i.e. vegetation) using an artificial neural network, see Figure 5. Each group of connected pixels classified as buildings is used as the ground plans for the building reconstruction.

After the data set has been classified, individual trees can be detected in the vegetation areas and the position, tree height, and crown diameter of the identified trees can be estimated (Persson et al, 2002, Hyyppä et al, 2001). In ongoing work we are also extending this method to perform tree species classification (Holmgren & Persson 2003).

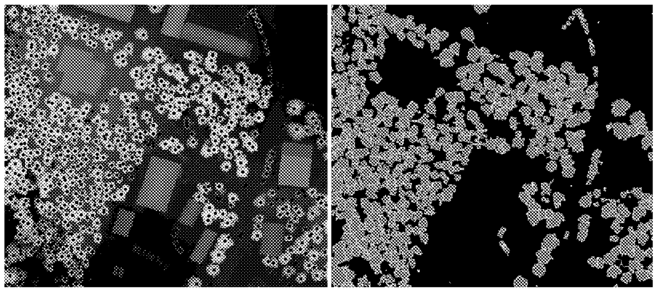


Figure 6. Estimated tree positions (left) and crown coverage (right).

In Figure 6, the result of tree detection is illustrated. The estimated positions of trees are marked on the elevation data and the obtained crown coverage is shown. For each identified tree, the height and crown diameter are also estimated. The maximum height value above the ground surface is chosen as the measure of the tree height. The area of the segments is used to calculate the crown diameter as if the crown has the shape of a circle. This data can then be used to populate the environment models with appropriately sized tree models.

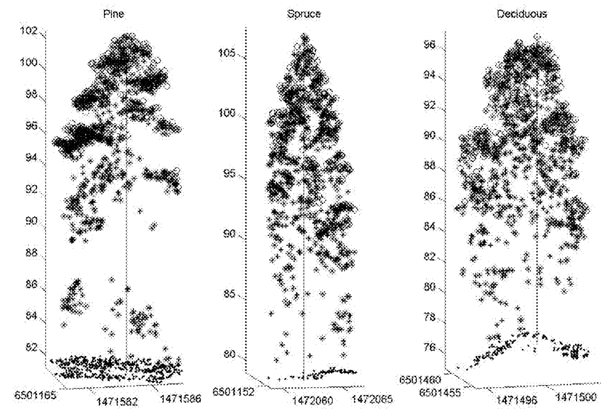


Figure 7. Sample laser data for a pine, spruce, and hardwood tree.

Using the segmented tree crowns, the tree species may also be classified. For this purpose we have recently developed a method for separating between spruce, pine, and hardwood trees. For each segment, all laser points within a crown segment can be grouped together to form a point cloud of the tree, see Figure 7. In order to separate spruce, pine, and hardwood trees, different features are extracted from the point clouds to capture variations in the crown shape and structure. The extracted features are based on geometry, intensity, proportions of laser returns of different types, and measurements of height distribution. In Figure 8 the classification result over a larger area is shown. Using manual inspection in high resolution colour images over the same area, the classified tree species correspond well with the visual interpretation of the digital images.

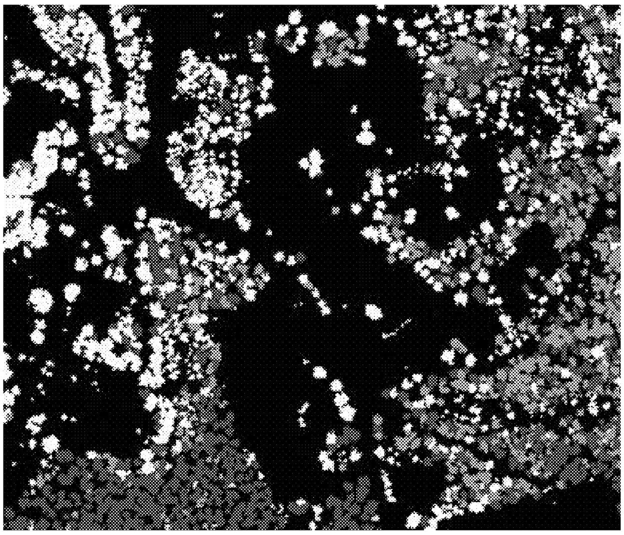


Figure 8. Result of tree species classification where dark grey represents pine, light grey represents spruce, and white represents hardwood trees.

3D Building reconstruction

3D reconstruction of buildings is an active area of research (Maas & Vosselman 1999, Haala & Brenner 1999, Vosselman & Dijkman 2001, Elaksher & Bethel 2002). In our work the objective is to develop a fully automatic method general enough to handle almost every building, even buildings having curved walls or domed roofs. For each ground plan detected in the classification, elevation data is used to extract planar roof faces. By following the outlines of roof faces, vertices are determined as well as intersection sections and height jump sections between adjacent faces. Having defined the relationship between the roof faces, a 3D polygon model of the building can be constructed.

Segmentation and extraction of planar roof faces

The planar faces of the roof are detected using clustering of surface normals. For each pixel within the building ground plan (Figure 9) a window is formed. The surface normal is estimated by fitting a plane to the elevation data within the window. The plane parameters are estimated using a least square adjustment. A clustering of the parameters of the surface normals is then performed. In the clustering space, the cell with the largest number of points is used as initial parameters of a plane and an initial estimation of what pixels belong to the plane is made using these parameter values. Finally, an enhanced estimation of the plane parameters using an iteratively re-weighted least square adjustment is performed to determine which pixels belong to the plane, (Zhang 1995). These pixels form a roof segment. This clustering process is repeated on the remaining non-segmented pixels within the ground plan, see Figure 10. Only roof segments having a certain minimum area are kept. After the clustering process, region growing is performed on the roof segments until all pixels within the ground plan belong to a roof segment.

Figure 9. Classified building ground plan.

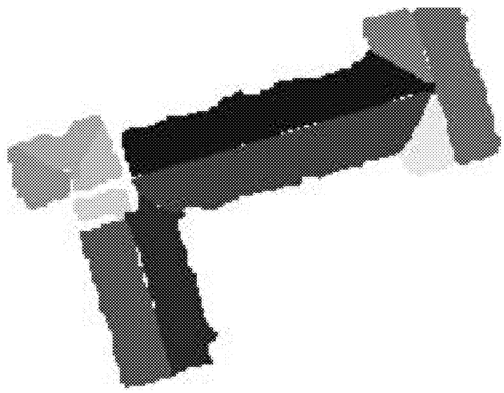


Figure 10. The roof segments.

Extraction of topological graph

Having segmented the roof faces of a building, the relationship between the faces is defined. By following the outlines of roof faces, so called 'topological points' are first added. Topological points are defined as vertices where a roof face's neighbour changes. In Figure 11, the topological points are marked with triangles. Next, each section of an outline between any two topological points is defined as either an intersection section, height jump section or both. For sections being defined as both an intersection section and jump section, a new topological point is added to split the section. The locations of the start and end points of the intersection sections are also adjusted according to where the roof faces intersect. In Figure 12, the intersection sections (bold), and the height jump sections are shown.

So far the height jump sections are defined by the chain code of the outline of the faces. Vertices along the height jump sections need to be estimated. As shown in Figure 12, the jump sections are noisy and difficult to model. Therefore, for each jump section, lines are estimated using the 2D Hough transform. The intersection points of the lines are used as vertices along the jump sections.

These vertices along the jump sections are used together with the topological points to generate a 3D polygonal model of the building.

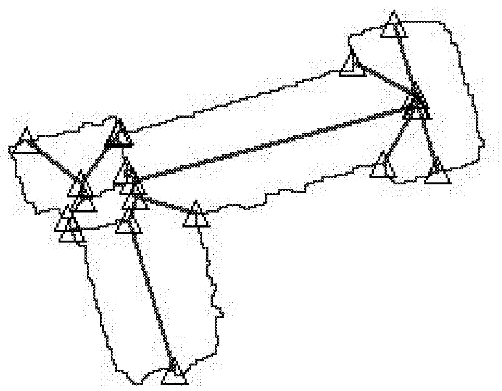


Figure 12. The topological points (triangles), the intersection sections (bold) and height jump sections of the building.

Generation of 3D models

After having defined the vertices of the roof faces and the relationship among these vertices and roof faces, a 3D polygon model can be created, see Figure 13 (bottom). Figure 13 (top) shows the 3D model when vertices along the jump sections are directly estimated without first estimating lines. The roof of the model is created using the extracted roof faces. The height value of the vertices is obtained from the z-value of the plane at the location of a vertex. Between any two vertices along jump sections, wall segments are inserted.

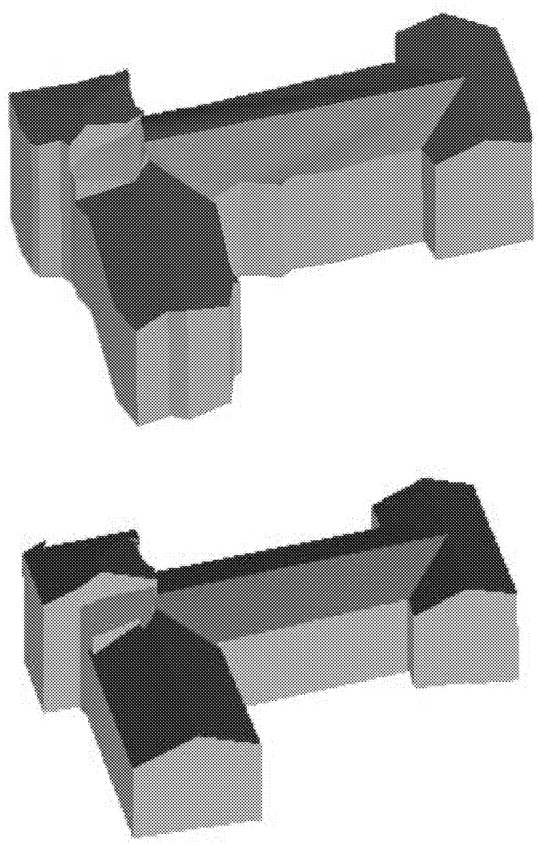


Figure 13. The 3D polygon model output from the general building reconstruction algorithm.

Environment modelling and visualisation

For creating environment models for e.g. decision support systems, training systems, etc the extracted features need to be integrated in some (automatic) modelling system. Prior to this, the extracted data are first exported in some suitable formats, e.g. vector files, 3D models or raster data

As an example, generating the environment model for the urban area in Figure 1 for a real time-visualisation application requires the bare earth DTM, an orthophoto mosaic, tree and building models and associated position and size information. The DTM is triangulated and draped with the imagery. The building models are placed in the correct positions and instances of 3D tree models are placed in the correct positions, all properly scaled and with the correct model for the detected tree species. This process is illustrated in Figure 14.

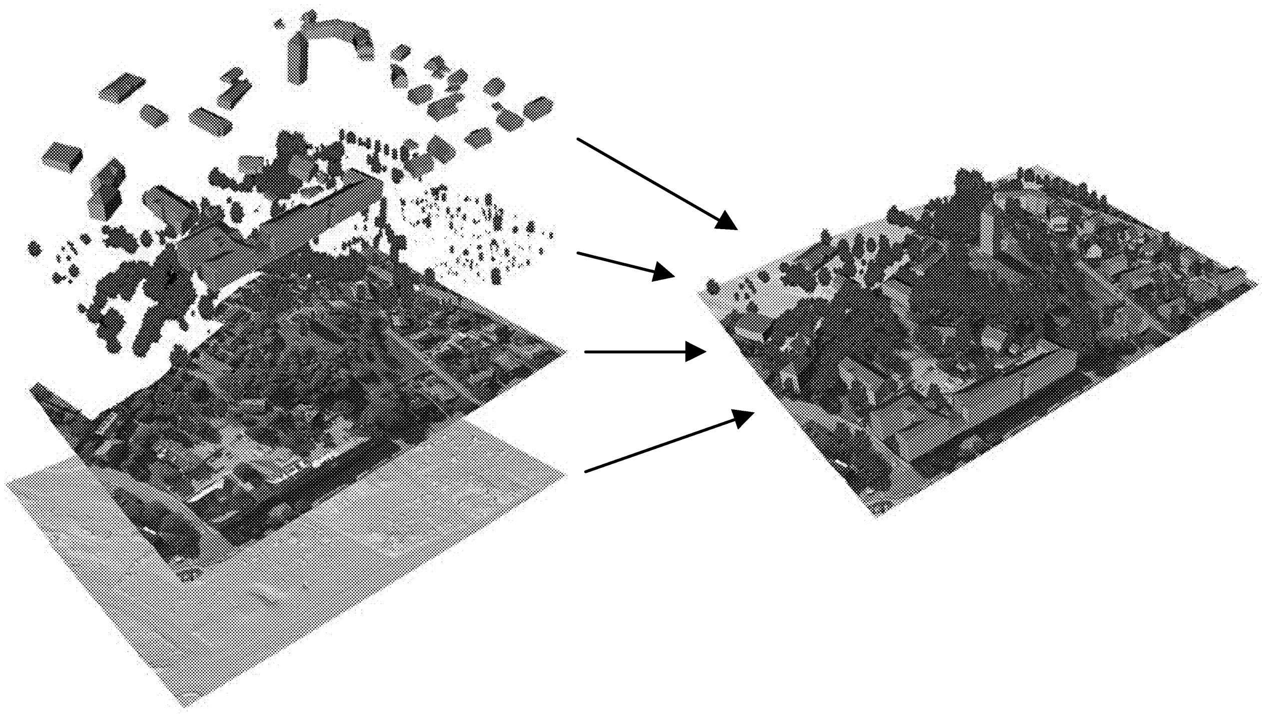


Figure 14. Extracted terrain features, an orthophoto mosaic and a DTM are integrated to create an environment model.

Conclusions and future work

There is an increasing demand for large and high-resolution environment models in many applications today and for this purpose automatic modelling methods need to be developed. In this paper we presented some steps toward a solution. We have shown that using high resolution data form airborne laser scanners (ALS) and digital cameras, automatic feature extraction and accurate 3D reconstruction of real world environments is possible. We have also shown how the result of feature extraction and 3D reconstruction can be used for the construction of high-resolution models for urban applications.

As examples of further work we see the need for using data from more sources, oblique texturing of buildings, further classification of terrain features and a more automated processes for application specific environment model generation.

References

Ahlberg, S. & Söderman, U. (2002). From Environment Data Collection to High Resolution Synthetic Natural Environments, *Interservice/Industry Training, Simulation and Education Conference*, Orlando, Florida, USA

Elaksher, A. & Bethel, J. (2002). Reconstructing 3D buildings from LIDAR data. *IAPRS volume XXXIV, part 3A, commission III, pp 102-107*.

Elmqvist M. (2002). Ground Surface Estimation from Airborne Laser Scanner Data Using Active Shape Models. *IAPRS volume XXXIV, part 3A, commission III,* pp. 114-118.

Kass, M., Witkin, A.,& Terzopoulos, D. (1998). Snakes: active contour models, *Int. J. of Computer Vision,* 1:321-331.

Haala, N. & Brenner, C. (1999). Extraction of buildings and trees in urban environments. *ISPRS Journal of Photogrammetry & Remote Sensing*, 54(2-3), pp. 130-137.

Holmgren, J. & Persson, Å. (2003) Identifying Species of Individual Trees Using Airborne Laser Scanner. Submitted to *Remote Sensing of the Environment*.

Hug, C. (1997). Extracting artificial surface objects from airborne laser scanner data. In Gruen, A, Baltsavias E.P. & Henricson O., eds, *Automatic Extraction of Man-Made Objects from Aerial and Space Images (II)*, Birkhäuser Verlag, Basel.

Hyyppä, J. Kello, O., Lehikoinen, M. & Inkinen, M. (2001). A segmentation-based method to retrieve stem volume estimates from 3-D tree models produced by laser scanners. *IEEE transactions on Geoscience and Remote Sensing*, 39(5), pp. 969-975.

Maas, H-G. (1999). Segmentation of Airborne Laser Scanner Data. Delft University of Technology.

Maas, H-G. & Vosselman, G. (1999). Two algorithms for Extracting Building Models from Raw Laser Altimetry Data. *ISPRS Journal of Photogrammetry and Remote Sensing 54(2-3*, pp. 153-163.

McKeown, D., Gifford, S., Polis, M., McMahill, J. & Hoffman, C. (1996). Progress in Automated Virtual World Construction. *Proceedings of the 1996 ARPA Image Understanding Workshop*, Palm Springs.

Persson, Å., Holmgen, J. & Söderman, U. (2002). Detecting and measuring individual trees using airborne laser scanning. *Photogrammetric Engineering and Remote Sensing*, 68(9), pp. 925-93.

Söderman, U. & Ahlberg, S. (2002). High Resolution Synthetic Natural Environments for Modeling and Simulation. *In the internet proceeding of the first Swedish-American Workshop on Modeling and Simulation* (SAWMAS2002), Oct 30-31 2002, Orlando, Florida. - *http://www.mind.foi.se/SAWMAS/SAWMAS-2002/index.html*

Vosselman, G. & Dijkman, S. (2001). 3D building model reconstruction from point clouds and ground plans. *IAPRS volume XXXIV, part 3/W4, commission III, pp 37-43*.

Zhang, Z. (1995). Parameter estimation techniques: A tutorial with application to conic fitting. Technical Report 2676, Institut national de recherché en informatique et en automatique, Unité de recherché INRIA, Sophia-Anipolis, France.

Distributed Simulation

Distributed Virtual Training Environment

Hassan Rajaei, Ph.D.

Dept. of Computer Science
Bowling Green State University
Bowling Green, Ohio, 43403
rajaei@cs.bgsu.edu

Abstract

This paper describes a Virtual Classroom for Training where students join the class and perform diverse tasks of the training agenda using distributed simulation and HLA, High Level Architecture. The virtual training environment is an interactive simulation based sphere distributed across a network or Web. The system is synchronous and allows participants to interact in real-time. The proposed method enhances the traditional Distance Leaning by utilizing distributed simulation technique as a key component of its virtual environment. The integrated distributed simulation and the synchronous real-time interactions in the virtual classroom facilitate furnishing a true interactive training atmosphere.

Keywords: Distance Learning, Distributed Simulation, HLA, Virtual Classroom, Virtual Environment

Introductions

Distributed Simulation plays an important roll in assessing ranges of applications in defense industry. One such application is training of defense personnel in a virtual environment. This research uses the same concept and attempts to adapt its principles to civilian educational domain. We call this environment Virtual Classroom for Training (VCT) that utilizes distributed simulation techniques in an interactive distance learning context. Contrast to the traditional distance learning, VCT utilizes distributed simulation as a key component of its virtual environment. Furthermore, the VCT distance learning is a synchronous environment whose activities are synchronized partly by the High Level Architecture (HLA) standard (Dahmann et. al. 1998, DMSO 1998) and partly by real-time digital video streaming.

Distance Learning (DL) is widely used today both in military and civilian institutes. Advances in the Internet technologies and e-learning provide numerous opportunities to furnish interactive services to defense and other industries, universities, civilian organizations and huge number of potential students scattered throughout the world. Example of efforts for this purpose includes the Department of Defense initiative on Learning Objective standards for Advance Distance Learning, ADL (ADL 2000) and Instructional Management Systems, IMS.

Most distance learning methods currently use Web-based technologies and multimedia for mainly disseminating their educational material. Our virtual classroom, however, enhances the distance learning approach by adding simulation techniques in a distributed environment. VCT attempts to follow ADL objectives to combine several technologies to provide a realistic training environment for the students participating in the classroom.

In this paper we describes the concept of our virtual classroom for training and identify its key components. Moreover, we provide the difference between our research and related work. In a follow-up paper we provide detail descriptions and simulation results of VCT as well as discussing some key issues. While this research is focused on educational domain, it could easily expand on other contexts such as industrial, healthcare, as well as back to the defense applications.

Virtual Classroom for Training

The heart of VCT systems consists of divers simulation objects, scenario libraries, data bases, as well as real objects. These components are especially integrated to facilitate training objectives. Participants of VCT could interact with simulated and non-simulated objects, with other trainees in the virtual training space, and with the instructor in a three dimensional graphical interface. In the virtual classroom, an instructor supervises the students by providing directives and supervisions. The instructor could be a real person, a simulated object, or a combination of the two. The students on the other hand are real and could share simulation objects with each other. In the following sections we describe more details of the system. Nevertheless, the descriptions provides only an overview of the system. More specific details will be provided in follow up papers.

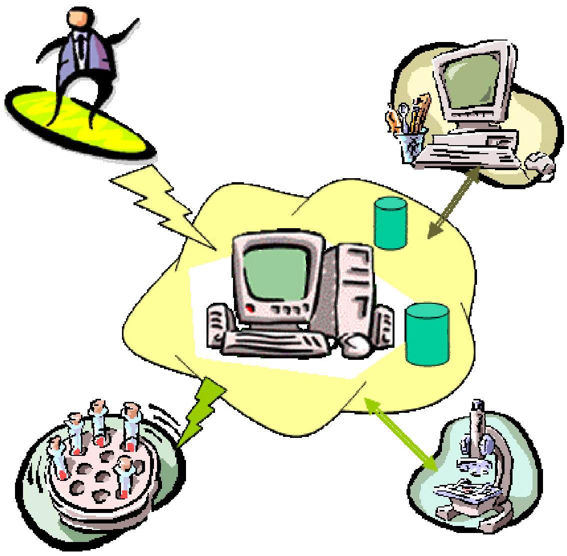


Figure 1: Example of a web-based Virtual Classroom enchanted by simulation which is suitable for asynchronous online Distance Learning.

Virtual Classroom

VCT provides synchronous services to heterogeneous platforms. The synchronous mode requires all students to join the class at a pre-scheduled time and for a specific duration. This is in contrast to the asynchronous mode where students connect to the system when they desire. Most Web-based online learning systems utilize the asynchronous approach an example of which is illustrated in Figure 1. The synchronous mode on the other hand is more complex, but provides better resource management, bandwidth supervision of the network, and hence furnishing better Quality of Service (QoS). Furthermore, it can ensure active participation of students in the educational program. We trust the synchronous mode can provide a true virtual classroom. Figure 2 illustrates an overall view of a synchronous Distributed Virtual Training Environment. In this system, it is assumed a Main Service Center, MSC, is the main controller who manages task distribution among available resources at its disposal. When parallel computing is needed, MSC could also co-ordinates that both at the local site as well as on remote nodes.

Participants can choose to join the classroom in three modes: physically entering the Main Service Center, logon to VCT via Intranet, or connect to the virtual classroom via Internet. While authentication is required for all participants, especial security checks required for those who connect to the system via Internet. Collaborative Service Centers (CSC) could also be used to augment the system and extend its outreach. However, this part will be addressed in future work.

The system uses XML based portals to disseminate information. Extensible 3D (X3D) is used for 3D presentation of graphical images. All participants are assumed to be equipped with special gloves, Head Mounted Display, one or more computer displays, broadband access to the network. The connection could be wireless or wire-based. All participants also need to have access to video conferencing equipments for audio/video real-time communications between the individual sites connected to the virtual classroom.

A training session starts by a training leader who provides the students the session agenda, theoretical lecture parts, and practical directives on the training assignments. A Virtual Assistant (Johnson et. al. 2003, Rizzo et. al. 2002, Shi et. al. 2003) is also available for all students as a help function. The VA could be programmed such that to alleviate part of instructor's work in order to minimize communications cost especially for the remote sites.

Depending on the length and type of the training sessions operation modes could be varied between lectures and laboratories. In the lecture mode, the students need not have to use the 3D headsets and gloves, whereas in the lab mode both are needed for embracing full visualization and interactions.

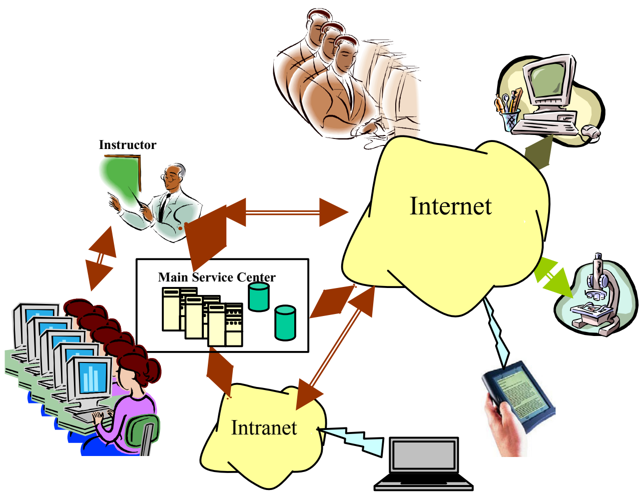


Figure 2: Overall System view of the synchronous Distributed Virtual Training Environment.

Virtual Classroom Objectives: Following objectives are pursued for virtual classrooms, whereas the ones for distributed simulation are addressed in the next section:

- Employ distributed simulation techniques as the core technology of the virtual training system
- Develop a Virtual Classroom Architecture flexible enough for both training purposes as well as traditional lecture oriented courses
- Pinpoint grand challenges and research on the key issues
- Collaborate with other scientists, institutes, and organization for the diverse issues involved in the development of such system.

Distributed Simulation

The core technology of VCT encapsulates Distributed Simulation based on High Level Architecture. Integration of HLA and Web-based simulation appears to be a natural pursuance for a Web application and distance learning (Morse 2000). The HLA framework provides a straightforward mapping of the distributed object entities into the virtual objects (Figure 3).

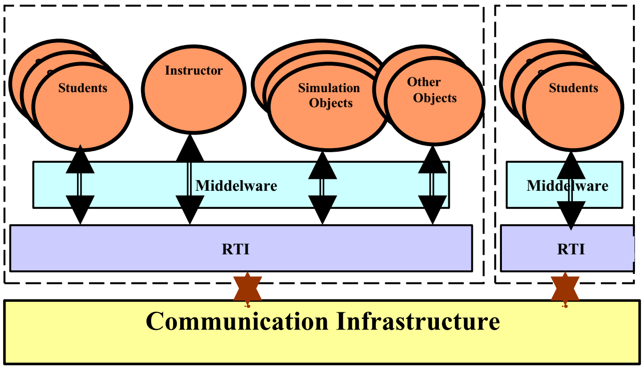


Figure 3: Overview of HLA Federations incorporating Distance Learning.

To facilitate interactions between diverse entities of the virtual environment, all components of a VCT system could be viewed as objects. This includes both simulation and non-simulation objects, for example images which are representing instructors, students, virtual assistants, and other components.

The Run Time Infrastructure (RTI) of HLA is used to synchronize the interactions of the distributed entities. It provides a common platform upon which both simulation and non-simulation objects could interact. A middleware platform is needed on top of RTI to facilitate integration.

The Local Time Warp (LTW) synchronization protocol (Rajaei et. al. 1993) could be applied for RTI. An overview of a distributed simulation system synchronized by LTW is illustrated in Figure 4. LTW maps each zone as a cluster. The activities within a cluster is synchronized by Time Ware protocol (Fujimoto and Weatherly 1996) whereas activities between the clusters are synchronized by a conservative protocol such as Conservative Time Windows, CTW, (Ayani, R. and Rajaei, H. 1994). The LTW model is extendible to more than two level of hierarchy. A third level can easily be built on top of the second level by relaxing its conservative constraints by employing protocols such as Breathing Time Bucket (Stienman 1991, Stienman 1993).

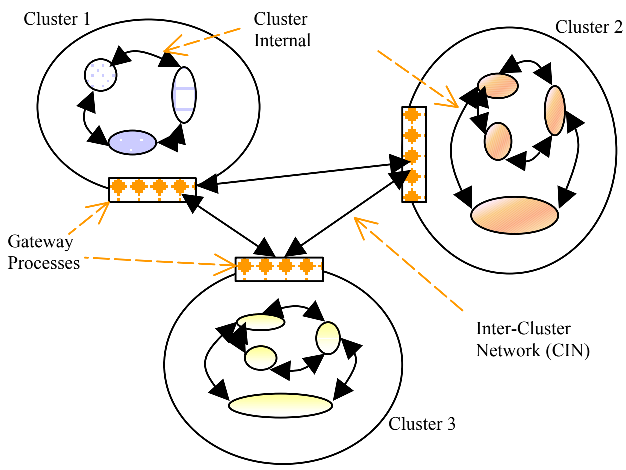


Figure 4: A synchronization mechanism for Distribute Simulation based on Local Time Warp clusters each zone in the global sphere.

Through HLA, the simulation model execution could be carried out on multiple sites where the RTI synchronizes the activities between different modules. If a single module is to be executed by e.g. the instructor, it could be executed on the MSC node and the result be multicast to the students. This suits well for large tasks when parallel and distribute computing is needed and MSC could accommodate the needed computing power. If MSC cannot accommodate the computing power for any reason, it acquires more resources from the distributed platform. If a simulation task for e.g. demo is simple enough and needs not significant processing power, then the execution will be carried out on each student's computer to avoid communications delay.

When a virtual interaction is needed to the images presented by the simulation executive, complications may arise depending on how many are engaging or viewing the result of the interaction in the virtual space. A typical case could exemplify for a collaborative group work where each student is in a different virtual site and yet all members of the group are working on the same simulated object.

Distributed Simulation Objectives: We pursue following key objectives for incorporating distributed simulation in the virtual training system:

- Develop a HLA compliant distributed simulation platform to suit Virtual Classroom objectives
- Research and develop an appropriate RTI and Middleware to support VCT
- Develop Library modules and scenarios appropriated for VCT and its future expansions
- Develop a test-bed to benchmark new proposals
- Focus on scalability and interoperability issues of the Distributed Virtual Training Environment.

Related Work

Success of HLA has pursued researchers to incorporate the standard into the Web. A successful attempt has been reported by (Morse 2000). Certainly the ADL Initiative has pursued numerous researchers to contribute their work in this important area. However, most efforts are focused towards military applications (e.g. Tolk 2003). The Extensible Modeling and Simulation Framework (XMSF) (Brutzman et. al. 2002) is believed to be suitable both for the civilian and military applications. Moreover, the framework is considered that it would overcome certain shortcomings of HLA over the Web.

Ever since the Internet becomes part of our daily work numerous techniques have been proposed for distance education and training (e.g. Shirmohammadi et. al. 2001, Zahariadis and Voliotis 2003, Alhalabi et. al. 2000). However, interactive distance leaning in real-time is much desired for a virtual classrooms (Kinser 2003). Achieving interactions in real-time is not trivial due to network delay, spite that several proposals have been emerged (e.g. Shi et. al. 2003, Moen and Pullen 2003).

Current virtual classrooms often use Web-based interactions by motivating students to answer questions through chat-rooms, video conferencing, whiteboard, or especial questionnaire. This model may work, but it is not scalable, nor true interactions could be achieved with the students. In contrast, we use simulation as a core component of our virtual classroom to motivate true interactions with the students. Albeit, this approach has been adapted by several virtual labs (Alhalabi et. al. 2000, Cracken et. al. 2003) and training environments (Crissey, et. el. 2002, Svensson et. el. 2002), our technique differs in using distributed simulation which is synchronized by HLA to incorporated scalability and interoperability.

Lara and Alfonseca (2003) incorporate visual interactive simulation for distance education as one core component of their system. There are two major differences with our work: their system is based on continuous simulation and not HLA compliant.

We trust that the High Level Architecture can provide an effective synchronization mechanism for the distributed virtual training environment where scalability of the system as well as interoperability of students' heterogeneous platforms comprise key concerns.

Challenges

VCT imposed numerous challenges. One of the grand challenges of the virtual classroom originates when the number of students increases in the system, especially when interactions between the participants themselves as well as with the simulation objects grows. The complexity increases when participants require real-time interactions with the simulated environment.

Network bandwidth and Quality of Service (QoS) imposes another grand challenge. Albeit Gilder's Law postulates that network bandwidth increases three times faster than CPU power, which doubles each 18 months (Moore's Law), in reality we are currently facing huge network capacity demand. This is certainly through when wireless links are used in the virtual sphere. Current quality of services for the wireless connections are very discouraging, especially for wide area networks . While the future most likely speaks for wireless connections, it is certainly a concern at present time.

Collaborative services though Collaborative Service Centers provides another complexity. We address this issue in the future work.

Figure 5: Collaborative Service Centers could provide virtual group work as if the members are present in a single place viewing and discussing same image.

Future Work

A true virtual classroom should integrate Virtual Presence of its participants in 3D images. The objective is that all participants should be able to feel they are in a single auditorium.

Collaborative environments for group work is most attractive and advantages, nevertheless much challenging, especially when interactions with the simulation images in the distributed virtual classrooms are to be presented to all participants. Several Collaborative Service Centers could jointly provide services to their respective zone. Moreover, distributed resources could be better managed for each zone, and yet share the resources with the virtual world. Figure 5 illustrate how a collaborative work could be done by virtual presence of the students who are participating in the virtual training session.

Various training and lecture may need varying length. Expansion to variable length of lecture and training can make the environment be suitable to mixture mode of training and lecture. That is, when the training is predominated then more simulation and interactions are anticipated, whereas a dominated lecture mode may need less interactions and simulation.

As stated in the related work, XMSF is believed to better suit distributed simulation in the Internet. We intend to research about this technique and conduct experiments to find about the feasibilities and capabilities. Moreover, we pursue to verify whether XMSF would suit better distributed simulation in the Internet.

Objectives of the future work:

- Provide proof of concept for VCT environment
- Expand the application of the generic architecture from the educational domain to cover other areas such as collaborative system development, healthcare simulation, and defense training
- Employ the emerging technologies for Advance Distance Learning.

Concluding Remarks

Advances in e-learning have paved the road for Advanced Distance Learning. Simulation could greatly enhance the learning process and furnish the needed active participation of students. Apparently numerous current distance learning systems have overlooked to this vital need of education.

We trust synchronous virtual classrooms enhanced by simulation technologies are more likely to embrace learning objectives by providing interactive learning sessions in a virtual educational sphere.

Embedded training and distributed simulation could become an integral part of the future Web-based distance education systems.

References

Advanced Distributed Learning Initiative., 2000. Shareable Courseware Object Reference Model, *www.adlnet.org*.

Alhalabi, B., Marcovitz, D., Hamza, K., and Hsu, S., (2000), Remote Labs: An Innovative Leap in Engineering Distance Education, *Florida Atlantic University, Center for Innovative Distance Education Technologies*.

Ayani, R., and Rajaei, H., Parallel Simulation Based on Conservative Time Windows: a performance study. Journal of Concurrency: Practice and Experience, Vol. 6(2) 119-142, April 1994.

Brutzman, D., Zyda, M., Pullen, M., Morse, K., (2002), XMSF 2002 Finding and recommendations Report, *Result of the Technical Challenges Workshop and Strategic Opportunities Symposium*, Oct 2002.

Cracken, S., Zilic. Z., and Chan, H., (2003). Real Laboratories for Distance Education, *CIT Journal of Computing and Information Technology,* Vol. 11, No. 1, 2003, 67-76.

Crissey, M., Thorstensson, M., Morin M, and Jenvald, J., (2002). How Modeling and Simulation Can Support MEDEVAC Training. *In Proceedings of the First Swedish-American Workshop on Modeling and Simulation* (SAWMASS'02), Oct 30-31, 2002, Orlando, Fl.

Dahmann, J., Fujimoto, R., Weatherly, R., (1998), *The DoD High Level Architecture: An Update, In Proceedings of IEEE/SCS Winter Simulation Conference*.

Defense Modeling and Simulation Office. (1998), High level Architecture Interface Specification, *hla.dmo.mil*

Fujimoto, R. M., and Weatherly, R. M. (1996), Time Management in the DoD High Level Architecture. *In Proceedings of the 10th Workshop on Parallel and Distributed Simulation*.

Johnson, W. L, Kole, S., Shaw, E., and Pain, H., (2003), *In Proceedings of Int. Conference on Artificial Intelligent in Education,*.

Kinser, K. (2003). Diversity Within the Virtual Classroom. *New Direction for Institutional Research, No. 118, Summer 2003, Wiley Periodicals, Inc.* 69-77.

Lara, J. and Alfonseca, M., (2003), Visual Interactive Simulation for Distance Education., *SIMULATION: Transactions of the Society for Modeling and Simulation International,* 2003.

Moen, D. and Pullen, M., (2003), Enabling Rea-Time Distributed Virtual Simulation over the Internet Using

Host-based Overlay Multicast., *In Proceedings of the IEEE/ACM Distributed Simulation Real-Time Applications Symposium,* 2003.

Morse, K. L., (2000), Taking HLA Education to the Wed. *In Proceedings of the 2000 Winter Simulation Conference,* 1619-1623.

Rajaei, H. Ayani, R., Thorelli, L. (1993), The Local Time Warp Approach to Parallel Simulation. *In Proceedings of the 7th Workshop on Parallel and Distributed Simulation*, 119-126.

Rizzo, P., Shaw, P., and Johnson, W. L. (2002), An Agent that helps Children to Author Rhetorically-Structured Digital Puppet Presentations, *In Proceedings of the Sixth International Conference on Intelligent Tutoring Systems*.

Shi, Y. Xie, W., Xu, G., Xiang, P., and Zhang, B. (2003). Project Smart Remote Classroom- Providing Novel Real-Time Interactive Distance Technologies. *International Journal of Distance Education Technology,* 1(3), 28-45, July-Sept 2003.

Shirmohammadi, S., Saddik, A., Georganas, N., and Steinmetz, R., (2001), Web-Based Multimedia Tools for Sharing Educational Resources. *ACM Jurnal of Educational Resources in Computing,* Vol. 1, No. 1, March 2001.

Stienman, J., (1991), SPEEDES: Synchronous Parallel Environment for Emulation and Discrete Event Simulation. *Advances in Parallel and Distributed Simulation, SCS Simulation Series* 23: 95-103.

Stienman, J, (1993), Breathing Time Warp. *In Proceedings of the 7th Workshop on Parallel and Distributed Simulation*, 109-118.

Svensson, E., Nahlinder, S., Danielsson, B., and Jenvald, J., (2002), Pedagogical Tools for Within Visual Range (WVR) Fighter Pilot Training - Another Step Forward. *In Proceedings of the First Swedish-American Workshop on Modeling and Simulation* (SAWMASS'02), Oct 30-31, 2002, Orlando, Fl.

Tolk, A., (2003), A Common Framework for Military M&S and C4I System. *In Proceedings of Spring Simulation Interoperability Workshop*, April 2003.

Zahariadis, T., and Voliotis, S., (2003), New Trends in Distance Learning Utilizing Net Generation Multimedia Networks, *Education and Information Technologies 8:1*, 67-81, 2003, Kluwer Academic Publisher, Netherlands.

Bandwidth Analysis of a Simulated Computer Network Running OTB

Juan J. Vargas, Ronald F. DeMara, Avelino J. Gonzalez, Michael Georgiopoulos

Department of Electrical and Computer Engineering
University of Central Florida
Orlando, FL 32816-2450
jvargas@ucf.edu, demara@mail.ucf.edu, michaelg@ucf.edu, gonzalez@ucf.edu

Abstract

A network simulation is conducted to determine the bandwidth requirements of military mission rehearsal activities while enroute to deployment. A predefined vignette running under OneSAF Testbed Baseline (OTB) provides the base data for the simulator. This data consists of Protocol Data Units (PDUs) captured by the OTB logger. The PDUs are the messages exchanged by all the entities in the simulation and include the timestamp and byte length, used in the simulation to recreate the network traffic.

The simulation includes 24 computers onboard 8 military airplanes, 3 computers per plane, representing 3 combat vehicles connected via a 100Mbps Ethernet LAN. A router interconnects the resources aboard each plane to those of other planes and also to a remote ground station via 64, 128, 256 and 1024 Kbps wireless communication channels. The results of the simulation are collected, plotted, and conclusions regarding bandwidth and latency implications of the Embedded Training exercises are drawn.

The embedded training computer network simulation is developed using the OMNeT++ public-domain modeling tool. OMNeT++ is a general discrete event simulation tool that contains features aimed to facilitate computer network modeling. It contains a graphical development environment used to define the topology of the network by instantiating and interconnecting different icons that represent the objects of the simulation, like computers, communication channels, routers, airplanes, etc. The behavior of each object is specified in C++. The final product is an executable standalone program that models the network behavior. It can run as an animation or as an express simulation. A wide range of statistics is obtained for analysis.

Introduction

In 2002, the U.S. Army Simulation, Training and Instrumentation Command (STRICOM) and the Computer Engineering Department of the School of Electrical Engineering and Computer Science at the University of Central Florida (UCF) began a joint project to assess the "Bandwidth and Latency Implications of Integrated Training and Tactical Communication Networks." The main goal of the project is to determine the bandwidth requirements for running Objective Force Embedded Training (OFET) methods.

OFET methods benefits include the ability to perform tasks such as mission planning and rehearsal while enroute to deployment. Benefits include the ability to perform "in-situs" exercises on actual equipment, more direct provision of support for the variety of equipment in the field, and a greater opportunity to develop new training exercises using much shorter lead times than were previously possible with stand-alone training systems.

A fully operational OFET platform also presents several technology challenges. In particular, management of Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) resources is required for successful integration of simulation within the actual environment. The purpose of this study is to assess communication requirements to support Enroute Mission Planning and Rehearsal (EMPR) in a Future Combat System (FCS) environment.

The first step towards the development of the network simulator was the design of the vignette illustrating a mission rehearsal operation enroute to deployment. This vignette included a database terrain, friendly and enemy forces, strategies, and other details representative of FCS operations. The vignette was run using the software simulator known as OneSAF Testbed Baseline (OTB).

Entities in OTB communicate to each other by passing messages. Messages in OTB are called Protocol Data Units (PDUs). PDU formats are part of the Distributed Interactive Simulation (DIS) protocol defined in the IEEE Standards 1278.1 (1995), 1278.2 (1995), 1278.3 (1996) and 1278.1a (1998). After running the OTB simulator, the PDUs generated in that session are collected and stored in an output file for further processing. Concerning the bandwidth project, the three most relevant features of the PDUs are the sender ID, the byte length and the timestamp of PDU creation because these parameters are the basic inputs for modeling communication traffic. The destination address is not considered given that OTB broadcasts all the messages to all the participating entities.

The developing team wanted a discrete simulation tool with the following characteristics: great flexibility from a programming point of view, Windows based for availability and readiness, Graphical User Interface (GUI) capable of showing simulation with animation, easy to learn for C++ users, and low cost. After a short survey, the team decided to use OMNeT++ a software tool developed by András Varga at the Budapest University of Technology and Economics (OMNeT++, 2003). OMNeT++ is a general discrete event simulation tool that contains features to facilitate computer network simulations. OMNeT++ runs under different platforms including UNIX systems, Linux and Windows. It contains a graphical development environment used to instantiate the entities of the simulation like computers, communication channels, routers, airplanes, etc. and connect them according to the desired topology.

Using OMNeT++

OMNeT++ is a discrete event simulation environment based on C++. It provides means for describing the topology of the network, either graphically or by using of the NED language (Varga, 2003). In particular, OMNeT++ operates on two types of files: NED and CPP.

The first type corresponds to NED files (xxx.ned) that describe the topology of the network composed of the different nodes involved in the simulation, their input and output gates and the connections between them. A NED file uses a special language called the NED language to describe modules. There are two types of modules: simple and compound. Simple modules contain no other modules inside them and are used to describe the most basic elements of the simulator. In this project, message generators, message sinks, communication channels (wireless and Ethernet buses) and routers correspond to simple modules. Compound modules contain other modules inside. For example, a computer onboard an airplane is a compound module because it contains a message generator and a sink. A plane is also a compound module that contains a computer, a router and an Ethernet bus. The largest compound module corresponds to the total network that contains airplanes, and a wireless bus to link the planes.

The second type of files corresponds to CPP files (xxx.cpp) that use C++ to describe the functionality of the simple modules. Each simple module requires a C++ source code that indicates how to process each packet that arrives to an input gate, as well as how to send a packet to an output gate. Using C++ code we can program the packet contents, its destination, its length, and the distribution used (exponential, normal, uniform, etc.) to schedule packets. The CPP files corresponding to the buses (Ethernet or wireless) handle the packet transmission and collision detection. Once the source files are ready, the simulator is compiled. In this project, Microsoft Visual C++ was used to accomplish this task. A special compiler supplied with OMNeT++ pre-processes the NED files creating cpp files. Next, all the cpp files are compiled together with the Tcl/Tk graphic library producing the executable file. Tcl/Tk is a graphic library in the public domain (Tcl Developer Xchange, 2003) and described in several books like (Zimmer, 1998). For C++ programmers, a sufficient understanding of the way OMNeT++ works along with the capability of developing basic models can be quickly achieved because the main concepts come from general knowledge about C++, as compared to other simulators in which there are several long manuals to read and the concepts are specific to that particular simulator.

Model Design

The model to be simulated is composed of eight airplanes carrying three computers and a router each, plus a satellite and a ground station. A random message generator with a specific distribution can be used to control the rate of packet generation. Alternatively, as in our study, each component actually reads the messages to be broadcasted from an input text file that contains the type, length and timestamp of each message as recorded by the OTB's logger when the vignette was run.

Module descriptions

The GNED editor is used to edit the NED files using a graphic interface. Figures 1, 2 and 3 show the OMNeT++ representation of the compound modules node, airplane and the whole network, respectively.

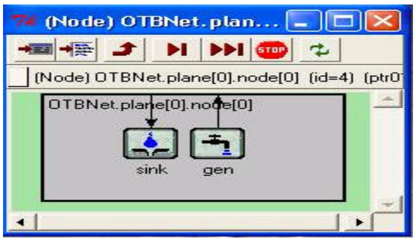


Figure 1. OMNeT++ view of a node containing a sink and a generator

Each node onboard an airplane contains a message generator and a message sink. The legend "`OTBNet.plane[0].node[0]`" in Figure 1 indicates that this compound module belongs to node # 0 located inside plane # 0 which is part of the general network called OTBNet. The planes are identified by using consecutive integers 0, 1, 2, …, 7. Within each plane, the computers are identified as nodes 0, 1 and 2. The brackets indicate that the set of planes and the computers onboard each plane are represented as arrays of simple modules. The arrows represent connections to input and output gates.

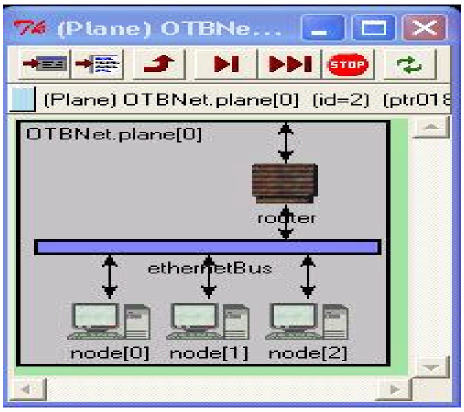


Figure 2. Airplane modules: embedded training stations, bus and router

Figure 2 shows the three nodes, the router and the Ethernet bus onboard each airplane. The three nodes and the router are connected via an Ethernet bus. Figure 3 shows the complete network composed of 8 airplanes, a ground station, a satellite and 3 wireless channels. The first wireless link connects the routers in all planes to each other. A second wireless link communicates the routers in the planes to the satellite, and the third one communicates the satellite to the ground station. In this way, each router is connected to three different links, and the satellite is connected to two.

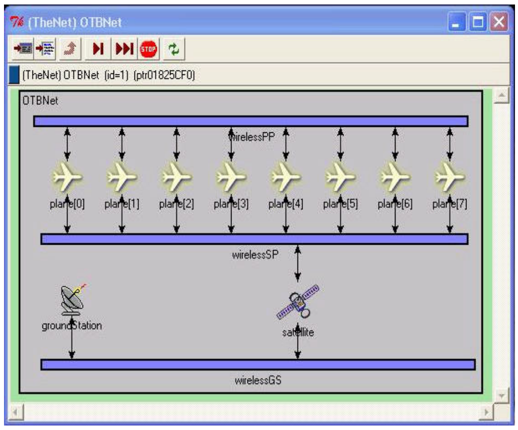


Figure 3. General view of the network showing 8 planes, satellite, ground station and the 3 wireless channels

Figure 4 contains the NED code of a generator, a sink and the satellite. For simple modules the NED code indicates the module name, the input parameters and the input and output gates. The C++ code of simple modules is written in a separate file. Each simple module becomes a C++ class.

```
simple Generator
parameters:
startTime: numeric,
fromAddr: numeric,
totalNodes: numeric;
gates:
out: out;
endsimple

simple Sink
gates:
in: in;
endsimple

simple Satellite
parameters:
startTime: numeric,
satelliteID : numeric,
satServiceTime : numeric,
totalNodes    : numeric,
WGSposition : numeric,
WSPposition : numeric;
gates:
in:  inBus1;
out: outBus1;
in:  inBus2;
out: outBus2;
endsimple
```

Figure 4. NED code of some simple modules

On the other hand, the NED code of a compound module includes additional features like the values of parameters of internal modules and connections between module gates. Compound modules do not need user-written C++

source code because their behavior is completely defined by their simple modules.

Input parameters of the model are used to control the conditions under which each simulation runs. These parameters can be directly given in the NED files, or they can be read from a configuration file at run time. Each simulation starts by reading the configuration file `OMNeT++pp.ini` containing initialization values.

Input Data

The input data to the model comes from the OTB logger. After setting up a particular vignette, it is simulated and the OTB logger records all of its PDUs into an output file that is later converted to text. OTB generates Persistent Object PDUs (PO_PDUs), which are a specialization of the general category of PDUs.

```
<dis204 po_variable PDU>:
dis_header.version=4
dis_header.exercise=1
dis_header.kind=250
dis_header.family=140
dis_header.timestamp=:01:33.432 (rel)
dis_header.sizeof=196
po_header.po_version=28
po_header.po_kind=2
po_header.exercise_id=1
po_header.database_id=1
po_header.length=147
po_header.pdu_count=7905
do_header.database_sequence_number=0
do_header.object_id.simulator.site=1082
do_header.object_id.simulator.host=23825
do_header.object_id.object=685
do_header.world_state_id.simulator.site=0
do_header.world_state_id.simulator.host=0
do_header.world_state_id.object=0
do_header.owner.site=1533
do_header.owner.host=23825
do_header.sequence_number=1
do_header.class=11
do_header.missing_from_world_state=0
reserved9=0
variable.total_length=132
variable.expanded_length=7812
variable.offset=0
variable.length=132
variable.obj_class=8
variable.data="Mine Pallet US M75
```

Figure 5. Example of a PO_PDU

Figure 5 is an example of a short PO_PDU. From among all the fields, the most important to our simulation are the site identification (1533), the length in bytes (147) and the timestamp (:01:33.432) interpreted as 1 minute, 33 seconds and 432 milliseconds. The timestamp represents the time the entity generated this PDU and put it on the output queue.

Simulation Results

The figures included in this article are labeled "Experiment # 3" because they correspond to the third experiment in a series of four. The experiment involved PDUs from six senders. The remaining 19 computers were listeners. The sites sending information were assigned to computers in separate airplanes and to the ground station. Two types of analyses were performed. The first one is a static analysis of the input data with no simulation involved. This analysis is subdivided into 2 categories: distribution of PDUs and minimum bandwidth requirements.

Distribution of PDUs

```
<dis204 fire PDU>:                  23
<dis204 po_objects_present PDU>:  682
<dis204 po_minefield PDU>:          14
<dis204 minefield PDU>:            117
<dis204 iff PDU>:                  851
<dis204 acknowledge PDU>:           36
<dis204 stop_freeze PDU>:            3
<dis204 po_line PDU>:              912
<dis204 po_task_authorization PDU>: 6
<dis204 po_task_frame PDU>:        382
<dis204 po_message PDU>:           119
<dis204 aggregate_state PDU>:      256
<dis204 po_delete_objects PDU>:    110
<dis204 po_parametric_input PDU>:1196
<dis204 laser PDU>:                  3
<dis204 detonation PDU>:            25
<dis204 po_fire_parameters PDU>:  713
<dis204 po_simulator_present PDU>:370
<dis204 entity_state PDU>:       28569
<dis204 mines PDU>:                386
<dis204 po_point PDU>:             659
<dis204 po_task_state PDU>:      11960
<dis204 signal PDU>:               237
<dis204 po_task PDU>:             2274
<dis204 po_unit PDU>:             1793
<dis204 transmitter PDU>:         8642
<dis204 start_resume PDU>:           3
                   Total PDUs = 60341
-------------------------------------
Site Assignment:
Site 1519 (50230 PDUs): plane=0, node=0
Site 1526 ( 1056 PDUs): plane=1, node=0
Site 1529 (  483 PDUs): plane=2, node=0
Site 1533 (  553 PDUs): plane=3, node=0
Site 1538 (  637 PDUs): plane=4, node=0
Site 1532 ( 7382 PDUs): ground station
```

Figure 6. Frequency distribution of PDUs

Figure 6 shows a frequency distribution of all the types of PDUs involved in the experiment, as well as the assignment of sites to model nodes. The assignment was based on the number of PDUs generated by each site, giving plane 0 the greatest number of PDUs and ground station the next greatest.

Minimum Bandwidth Requirements

The PDUs of all the sites were merged into one single stream of data and sorted according to their timestamps prior to any bandwidth computation. This was done because in the DIS protocol all the PDUs are broadcasted and any listening site will have to receive the PDUs from all the generating sites as one single stream of data. Then, without using simulation, a separate program calculated the minimum instant bandwidths by dividing the total simulation time into small time intervals of 2 seconds each, and computing the ratio of volume of data transmitted in each interval to the length of the interval.

Figure 7 shows the graph of minimum bandwidth requirements at each time interval. PDUs indicate that the activity starts at second 1035 and ends at second 2550 for a total time of 25 minutes and 15 seconds of simulation time.

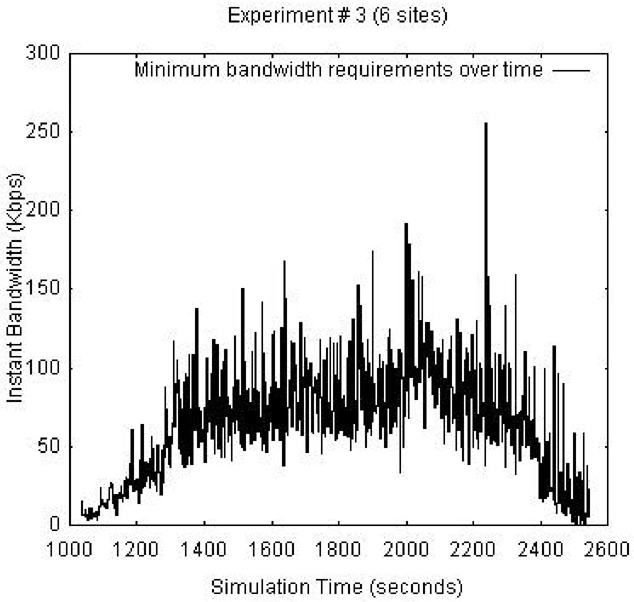


Figure 7. Minimum bandwidth requirements

In the static analysis, overheads like retransmissions, packet losses, or collisions are not considered. Therefore, the resulting bandwidth estimates can be interpreted as an absolute lower bound for the actual required bandwidth. Because a time gap must exist between packets as indicated by the IEEE Standard 802.11 (1997), it was set to 50 microseconds in this analysis. From the graph in Figure 7, the static analysis indicates that the maximum required bandwidth is near 256 Kbps, but the majority of the time the required bandwidth is less than 200 Kbps.

The second analysis performed corresponds to results obtained by running the simulator. The analysis is subdivided into 4 categories: slack time analysis, travel time analysis, queue length analysis and collision analysis.

Slack Time Analysis

The slack time for each node generator is defined as the difference between the timestamp of each PDU and the current simulator time at the moment the PDU is read from the input file. If the difference is positive then the generator is ahead of the planned schedule, otherwise it is behind it. Thus, a negative slack time indicates that the channel bandwidth is not enough to transmit the required PDUs without delay.

Figure 8 shows the slack time for all the units (routers and ground station) when the wireless channels are set to 64 Kbps

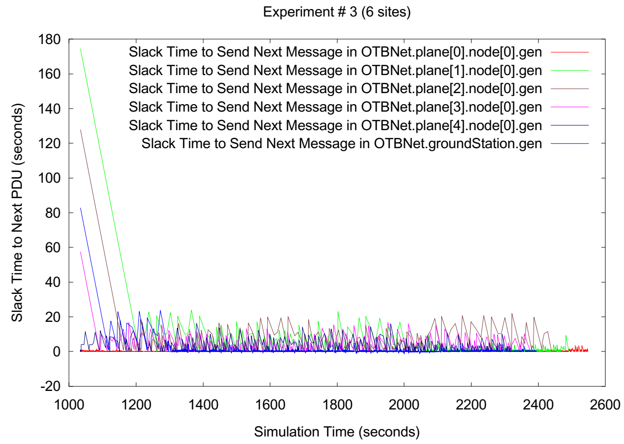


Figure 8. Slack times at each transmitting node using 64 Kbps in the wireless channels

Although Figure 8 gives the impression that no negative slacks are produced, a close up near zero in the Y axis reveals that many negative spikes do exist, most of them generated by the ground station. This is explained by the fact that the generators onboard the planes are directly connected to high speed Ethernet buses (100 Mbps), while the ground station is connected to a low speed wireless channel (64 Kbps).

Figure 9 shows a close up of the slack time for the ground station at 64 Kbps. An increase in the wireless bandwidth decreases the negative spikes, but does not eliminate them completely. One contributing factor to the negative spikes is the fact that during certain events OTB produces sequences of PDUs having exactly the same timestamp. For instance, sequences of 8 or more "`po_fire_parameters`" PDUs having the same timestamp were detected. Each PDU requires some positive transmission time, and so when the next PDU arrives its timestamp is previous to the current time and a negative spike starts to build. Further research is being conducted to analyze the composition of the PDUs participating in the negative spikes and to propose ways to eliminate these spikes. One possibility is to group together PDUs of the same type and length into one single packet.

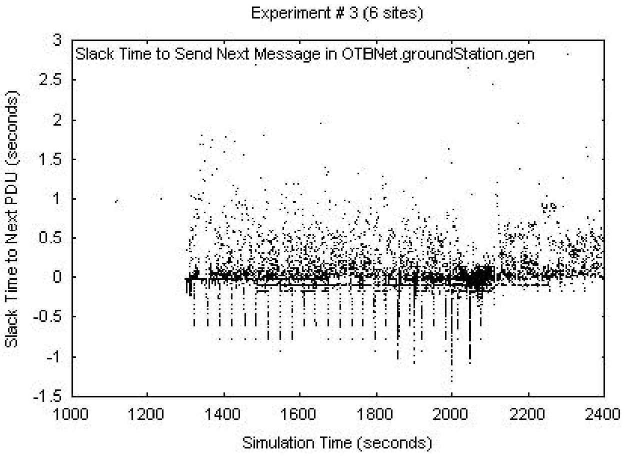


Figure 9. Slack time at ground station using a bandwidth of 64 Kbps

Travel Time Analysis

The travel time is the difference between the sending time of a PDU from a node generator and the arrival time to a node sink. All the transmission times, propagation times and waiting times in router queues are part of the travel time. Figure 10 shows the travel time of PDUs measured by the sink at node 0, plane 0, using 64 Kbps on the wireless links.

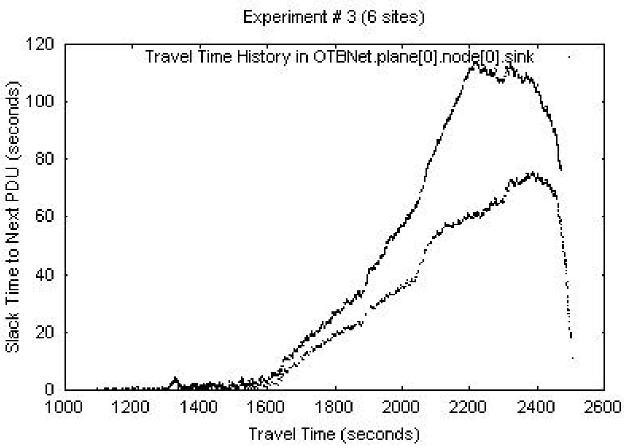


Figure 10. Travel times measured at node 0, plane 0

At node 0 the graph clearly shows two traces corresponding to two types of PDUs. The PDUs that take longer to arrive come from the ground station. These PDUs had to wait on the satellite queue as well as on the router queue. On the other hand, the PDUs coming from computers onboard the other planes had to wait on the router queue only. As seen, at 64 Kbps the travel times of most PDUs are completely unacceptable. Some PDUs took more than 100 seconds since the time they were sent to the time they arrived.

Although not shown, the ground station presents a similar behavior due to the relatively long queue and transit times associated with the satellite. An increase in the bandwidth produces a significant reduction in travel times. At 200 Kbps, the travel times to the ground station are less than 1 second. Considering that the minimum travel time is about 0.25 seconds due to the distance from the satellite to Earth, latencies less than 1 second are within the same order of magnitude from being optimal. At 200 Kbps, a better packet scheduling policy could diminish the negative spikes, especially if OTB were set to deliver the PDUs in a less burst mode, which is also the topic of future research.

Queue Length Analysis

There is a message queue at the satellite and at each router. Every time a PDU arrives to a router or satellite, the number of messages in the system is counted including the just arrived PDU, the PDUs in the queue and any one being serviced, and the tally is recorder along with the arrival time.

The two most important queues to analyze are the queues at the router onboard plane 0 and at the satellite. Figure 11 shows the messages in the router queue, using 64 Kbps in the wireless channels.

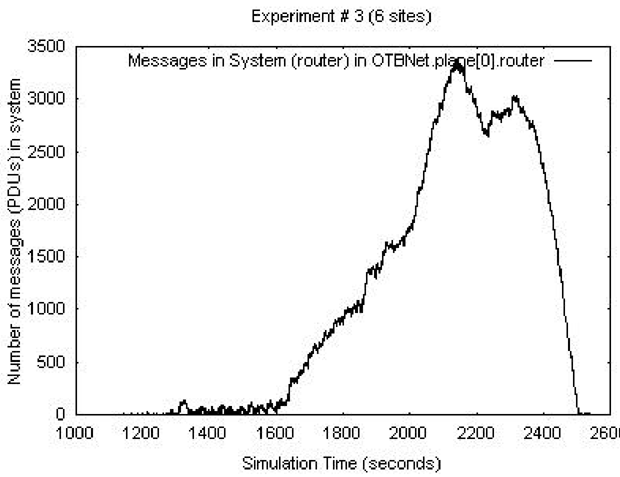


Figure 11. Messages in router queue at plane 0

As observed, the queue length becomes really unacceptable, reaching more than 3000 messages during the worst periods. Similarly, the queue at the satellite has a peak of more than 2200 messages. On the other hand, routers at other planes have acceptable queues. For instance, plane 3 (graph not shown) has a queue with a maximum of 23 messages. The reason is that the corresponding node transmits only 553 PDUs, a number easily handled by the router.

Simulation runs using bandwidths of 64, 200, 512 and 1024 Kbps in the wireless channels showed the effect of a bandwidth increase over the queue length in routers and in the satellite. The results indicated that just by increasing the bandwidth to 200 Kbps, the queue length decreases to less than 60 and less than 40 messages at plane 0 and at the satellite, respectively, in the highest peaks, which is quite significant.

Collision Analysis

The satellite and routers keep separate counters of collisions on each of the buses they are connected to. The satellite and the routers are connected to 2 and 3 links, respectively, as explained in the Model Design section. Each time a collision is detected, the corresponding counter is updated and its value along with the current simulation time is recorder for future processing. OMNeT++ includes procedures to collect and plot such information. The buses were programmed in such a way that packets transmitted by a module are not returned to its sender, even if they collide. Therefore, active senders are not good candidates to monitor collisions. In this analysis, the router at plane 7 was chosen to count collisions because none of the nodes onboard plane 7 is an active sender, making the router a good indicator.

The simulation results at 64 Kbps indicate that the highest collision rate occurred in the bus connecting the satellite to the planes and was close to 12 collisions per second. More than 6000 collisions were detected at 64 Kbps, which represents approximately 10 % of the total number of PDUs. At 200 Kbps, the router at plane 7 detected a maximum of 4 collisions per second. At this bandwidth, the total number of collisions was near 4500, or 7% of all the PDUs. Collision accumulation in plane 7 at different bandwidth rates is given in Figure 12.

The collisions were calculated as the sum of collisions detected in the three buses the router is connected to. However, the main component comes from the wireless link between the airplane and the satellite. The behavior of this link depends on the assignment of OTB transmitting sites to computer nodes. For example, if the PDUs currently assigned to the ground station were assigned to computer node 1 onboard plane 0, the Ethernet link running at 100 Mbps will produce fewer collisions than a wireless link running at 64 Kbps due to the shorter transmission times.

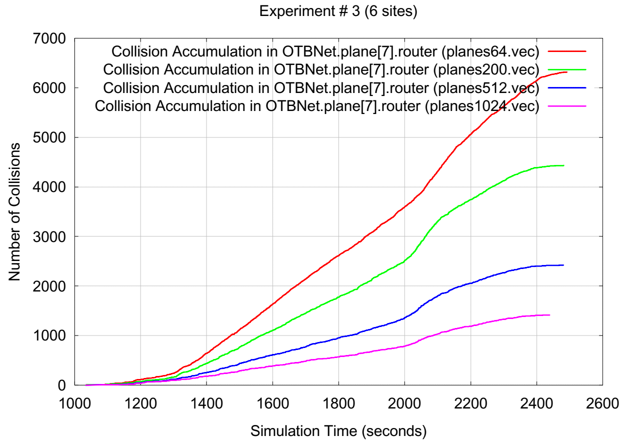


Figure 12. Collision accumulation at plane 7 at 64, 200, 512 and 1024 Kbps

Table 1 shows statistics about the total number of collisions, percentage of all the PDUs, and average number of collisions per second at different bandwidths.

Table 1. Collision accumulation, percentage and average number of collisions per second at different bandwidths

64KBps	6300 coll.	10 %	2.5 coll/sec
200KBps	4400 coll.	7.3 %	1.7 coll/sec
512KBps	2300 coll.	3.8 %	0.9 coll/sec
1024KBps	1300 coll.	2.1 %	0.5 coll/sec

Conclusions

As predicted by the static analysis, a bandwidth of 64 Kbps in the wireless links is insufficient to handle embedded training traffic under a DIS protocol. Latencies of more 100 seconds were detected for traffic coming from the ground station where the simulated Opposing Force resides. A significant improvement was achieved at 200 Kbps, where latencies less than 1 second were almost always the case for messages received at the ground station.

At the router in plane 0 and at the satellite, the queue lengths changed from 3400 and 2200 (max. peak) to less than 60 and less than 40 (max. peak) messages, just by increasing the bandwidth from 64 to 200 Kbps. As seen by the router at plane 7, collisions decrease and become manageable as the bandwidth increases.

Regarding modeling tool used, OMNeT++ is easy to learn and versatile to use for modeling computer networks. Several network models have been built that are available for downloading from the Internet. These models can be tailored to the user needs. OMNeT++ offers the possibility of running the simulation at several speeds with or without animation, including faster-than-real time. The results can be displayed and plotted on the screen as the simulation progresses, and also can be stored into text files for further processing.

References

IEEE (1995). *Std 1278.1. Standard for Distributed Interactive Simulation — Application Protocols*. IEEE Computer Society Press.

IEEE (1995). *Std 1278.2. Standard for Distributed Interactive Simulation — Communication Services and Profiles*. IEEE Computer Society Press.

IEEE (1996). Std 1278.3. *Recommended Practice for Distributed Interactive Simulation — Exercise Management and Feedback*. IEEE Computer Society Press.

IEEE (1997). *Std 802.11 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*. IEEE Computer Society Press.

IEEE (1998). Std 1278.1a. *Standard for Distributed Interactive Simulation — Application Protocols*. IEEE Computer Society Press.

Tcl Developer Xchange (2003). Retrieved November 15, 2003 from http://www.tcl.tk/.

Varga, A. (2001). OMNeT++ Discrete Event Simulation System. In *Proceedings of the European Simulation Multiconference* (ESM'2001). June 6-9, 2001. Prague, Czech Republic. User manual retrieved December 1, 2003 from http://www.omnetpp.org/index.php.

Varga, A. (2002). OMNeT++. In the column *"Software Tools for Networking", IEEE Network Interactive*. July 2002, Vol. 16 No. 4.

Zimmer, J. (1998). *Tcl/Tk for Programmers*. IEEE Computer Society Press.

A Study of Cluster Computing over IEEE 1394

Catheryne Williamson[1], Daniel Williamson[1], Janusz Zalewski[2]

[1] School of Electrical Engineering & Computer Science
University of Central Florida, Orlando, FL 32816
[2] Computer Science Program
Florida Gulf Coast University, Ft. Myers, FL 33965

Abstract

This study investigates the effectiveness of using IEEE 1394 serial bus under Linux for Cluster Computing in a view of achieving higher performance of real-time simulation. Studying the benefit of isochronous transmission for various devices and network configurations, as well as workloads, was a particular objective of this research. In this view the effect of the use of multicasting on improvement or degradation of performance has been also analyzed. The tests have been performed using sockets and MPI implementations to compare IEEE 1394 and Fast Ethernet. The results show the effectiveness of using isochronous mode over asynchronous mode and the use of multicasting over broadcasting. The study also shows how the network can be optimized in different ways and suggests continued research that can be done to further investigate the performance of the IEEE 1394 under Linux for Cluster Computing.

Introduction

This study investigates the effectiveness of using IEEE 1394 serial bus under Linux for Cluster Computing. Computations that require a lot of computing power, such as simulations of atmospheric conditions, ocean currents, nuclear reactions, etc., have traditionally been done by supercomputers. This has been very expensive and complicated in terms of programming effort. With the advent of very cheap PCs which can be connected into computer clusters running under a free operating system Linux, small companies or universities can build their own networks which have the computational power of supercomputers for a relatively small investment. However, due to unpredictability of computer networks, new problems arise, such as communication delays, packet losses, channel failures, etc.

One of the focal points of building inexpensive computing clusters is to use an appropriate interconnect technology. In this project, IEEE 1394 serial bus (also known as FireWire) is used for this purpose, due to its ability to provide guarantees for real-time traffic via the use of isochronous transmission mode. Studying the benefit of isochronous transmission for various devices and network configurations, as well as workloads, is the primary objective of this research. Secondly, for any type of interconnects it is crucial to use appropriate protocols to optimize traffic. In particular, effect of the use of multicasting on performance is important and constitutes another objective. The tests have been performed using sockets and MPI implementations to compare IEEE 1394 and Fast Ethernet.

IEEE 1394 Serial Bus

IEEE 1394 serial bus (IEEE, 1995) has been designed to accommodate fast data transfer (max. 400 Mb/sec.) among computing nodes connected via a serial cable (up to approx. 70m). The standard supports two different types of transfer modes between nodes. The transfer which requires periodic data transmission without guaranteed data delivery is called asynchronous transfer, while the kind of data transfer that is crucial to other applications and requires guaranteed delivery is called *isochronous*.

The asynchronous protocol transfers the data and transaction layer information to an explicit address. The isochronous protocol broadcasts the data based on channel numbers. This feature of supporting both protocols (asynchronous and isochronous) on the same interface makes both non- real time and real-time traffic possible.

The isochronous communication guarantees a particular time slice each 125us (8,000 isochronous cycles per second) dedicated for real-time traffic. Since a real-time device is guaranteed a time slot and isochronous communication takes over asynchronous, isochronous bandwidth is assured. Ongoing isochronous communication between two or more devices is referred to as a channel.

The delivery of the data in Isochronus Transfer Mode does not require confirmation of the delivery as in asynchronous transfer, but the data have to be delivered at a constant rate across the bus. Serial bus data transaction takes place via a series of packet transfers. Before a transaction begins, the requestors initiate the transaction on the target device, which calls the responder to receive the request.

It is important to mention that isochronous transfer defines a channel number rather than a unique physical address. This makes isochronous transfer broadcast over the bus and reach one or more devices based on the channel number associated with this transfer. These kinds of transfers require immediate bus access and therefore isochronous transfers have higher priority than asynchronous transfers. Every node that wants to do isochronous transfers has to make a request for required bandwidth. The request is sent to the node that performs the role of the isochronous resource manager.

There are four protocol layers defined for IEEE 1394: Bus Management Layer, Transaction Layer, Link Layer, and Physical Layer. Each of these layers provides a set of services to support communications between the applications running on the IEEE 1394 nodes.

There are several different functions that the Bus Management Layer supports. Every node must include support for automatic bus configuration, while other bus management functions are optional. Besides this some nodes participate in the global bus management to ensure that the nodes on the bus performs their functions efficiently. The global bus management consists of the following:

- Allocating channel numbers and bus bandwidth for isochronous transfers.
- Controlling the intervals at which isochronous transactions are performed.
- Verification that all bus powered nodes have power.
- Providing services to other nodes (e.g. specifying the maximum speed at which two nodes can communicate with each other).

The Transaction Layer provides services for asynchronous transaction types: Read, Write and Lock. The Transaction Layer does not provide any services to the isochronous transactions.

The Link Layer provides the translation of the Transaction Layer's request or response into a corresponding packet to be sent over the serial bus. It is also is responsible for sending and receiving the isochronous data.

The Physical Layer provides the electrical and mechanical interface between the Link Layer and the serial bus. This layer also provides an arbitration process that insures only one node at a time is transferring data.

Experimental Setup

The research question for us was if and how much the isochronous mode is superior to traditional ways of transmitting data during computations, such as asynchronous mode, Ethernet and TCP/IP protocol. For this purpose, the testing setup was built up of three computers and three IEEE 1394 cards as shown in Figure 1. Their description is summarized below.

Kalman and Coriolis. SMP architecture with two 731MHz Pentium III processors and 512 MB of RAM with 256 KB cache; local disk drive 17.9 GB SCSI Seagate ST318436LW; Linux kernel 2.4.2 with Mandrake 7.2 distribution for Kalman and Red Hat 7.0 distribution for Coriolis.

Frick. Single 75 MHz Pentium I processor with 29.9 MB RAM; local disk drive 1 GB; Linux Kernel 2.4.2 with Red Hat 7.0 distribution.

All three computers have IEEE 1394 PCI Host Adapter boards from Texas Instruments. The 1394 to PCI chip is TSB12LV21APGF (PCILynx) and 1394 to PHY chip is IBM21S850PFD.

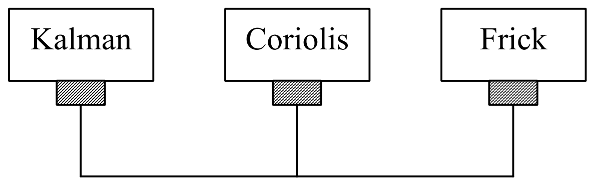


Fig. 1. Computer configuration for experiments.

Testing Experiments

Two basic types of experiments were conducted: tests for isochronous mode and tests for multicasting. Results of both tests are discussed below.

Tests for Isochronous Mode

Comparison of Different Protocols. This comparison was done for the IP-based driver (ip1394) and Ethernet-based driver (eth1394) to see which one is faster and better choice to use for IP transfers. The tests are generated using NetPIPE (1997). The Maximum Transfer Units (MTU) values in the drivers are changed to see the effect on performance.

Figures 2 and 3 show performance for both ip1394 and eth1394 driver with different MTU settings. Maximum values defined are 2030 for both drivers. The default values are 124 for ip1394 and 898 for eth1394. The smallest values are 124 for ip1394 and 186 for eth1394.

Figure 2 shows transmission rate in Mbits/s. The maximum transfer rate obtained with maximum MTU value of 2030 for ip1394, is 169 Mbits/s or 21.1 MB/s.

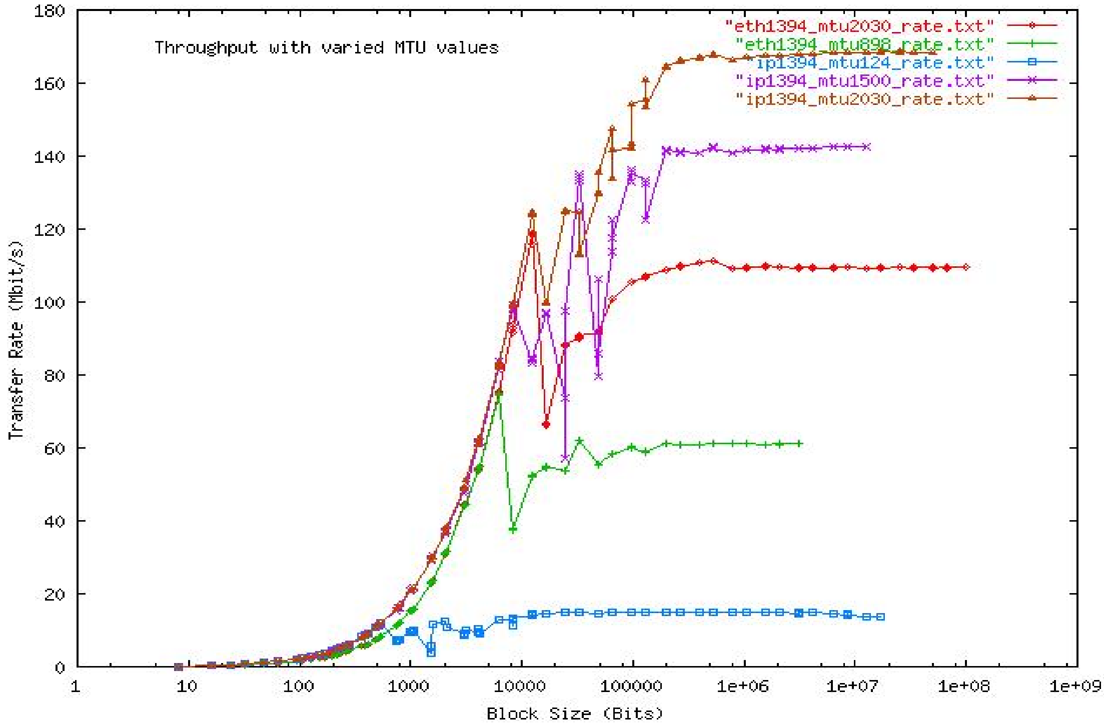


Fig. 2. IEEE 1395 throughput for varied MTU values.

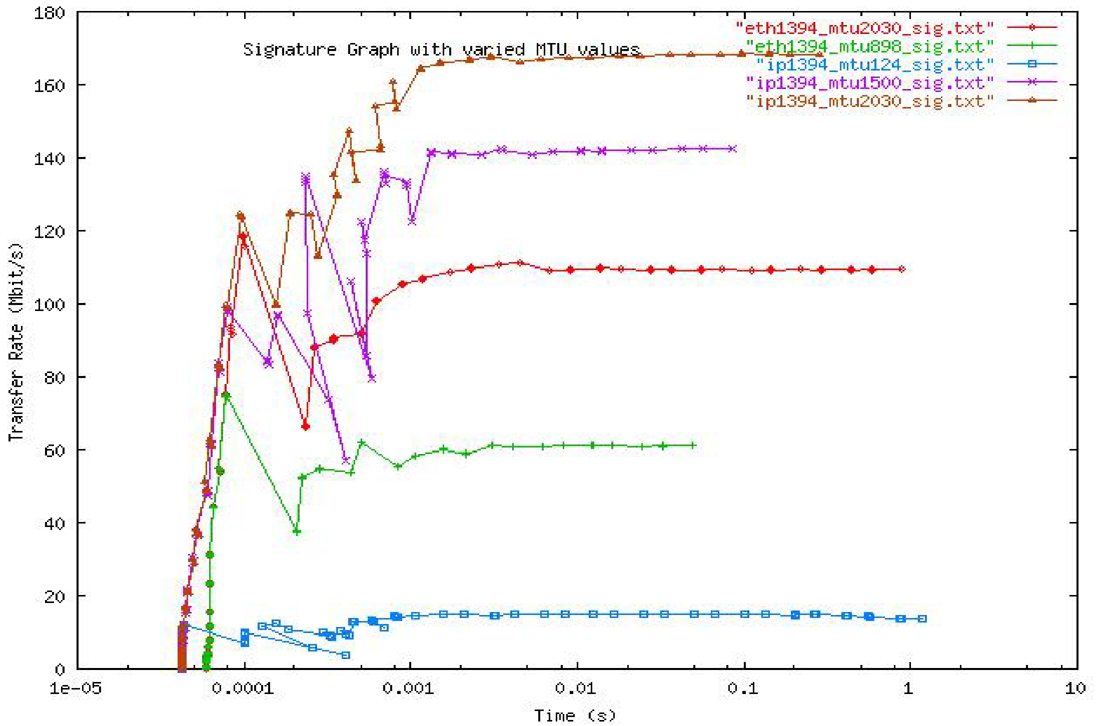


Fig. 3. IEEE 1394 signature for varied MTU values.

For eth1394 the maximum transfer rate was also reached at the maximum MTU value of 2030, and equals 116 Mbits/s. The maximum value theoretically attainable with the IEEE1394 standard is 400 Mb/s or 50 MB/s. Another important conclusion that can be drawn from reading this graph is the big difference the MTU values make. If the default values of 124 for ip1394 and 898 for eth1394 were used, not only would the eth1394 have the highest transfer rate, but also the difference for both drivers would be significant.

On the signature graph in Figure 3, the maximum transfer rates can be read more easily. Here the *acceleration* can also be evaluated between the starting point and the maximum throughput. For the curve with the maximum throughput (ip1394 with MTU 2030) this acceleration is (169-0) / (0.002-0.00003) = 85787 bits/s². In this experiment the latency values are between 220µs-340µs.

Raw driver. To test the isochronous transmission using the raw1394 driver an application was written for both asynchronous and isochronous transmission. These testing benchmarks have been developed to show the same types of graphs as NetPIPE, but without having to use the IP protocol.

The number of isochronous cycles is fixed. The maximum packet size allowed to be sent depends on the speed. Therefore the test involves sending different packet sizes in each isochronous cycle.

One limiting factor in this case is that the PCILynx card allows transmission only on one channel at a time. The PCILynx hardware does not support multiple channels as the OHCI card does (Anderson, 1998). Multiple channel transmission would increase the transfer rate.

Figure 4 shows the throughput comparing isochronous transfer to the asynchronous transfer. The maximum transfer rate is 163 Mb/s for isochronous transfer and 75 Mb/s for asynchronous transfer.

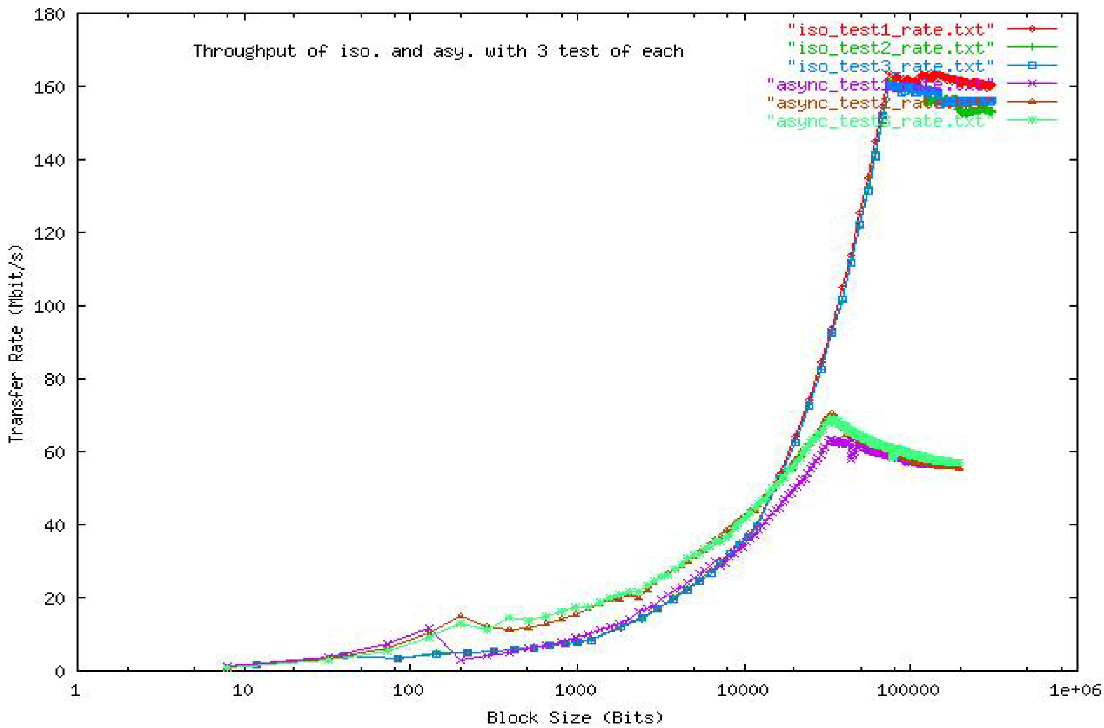


Fig. 4. IEEE 1394 throughput over raw driver.

On the signature graph for raw1394 (Fig. 5) the acceleration is not as evenly spread as on the ip1394/eth1394 signature graph. The latency is however about the same, 210µs-340µs.

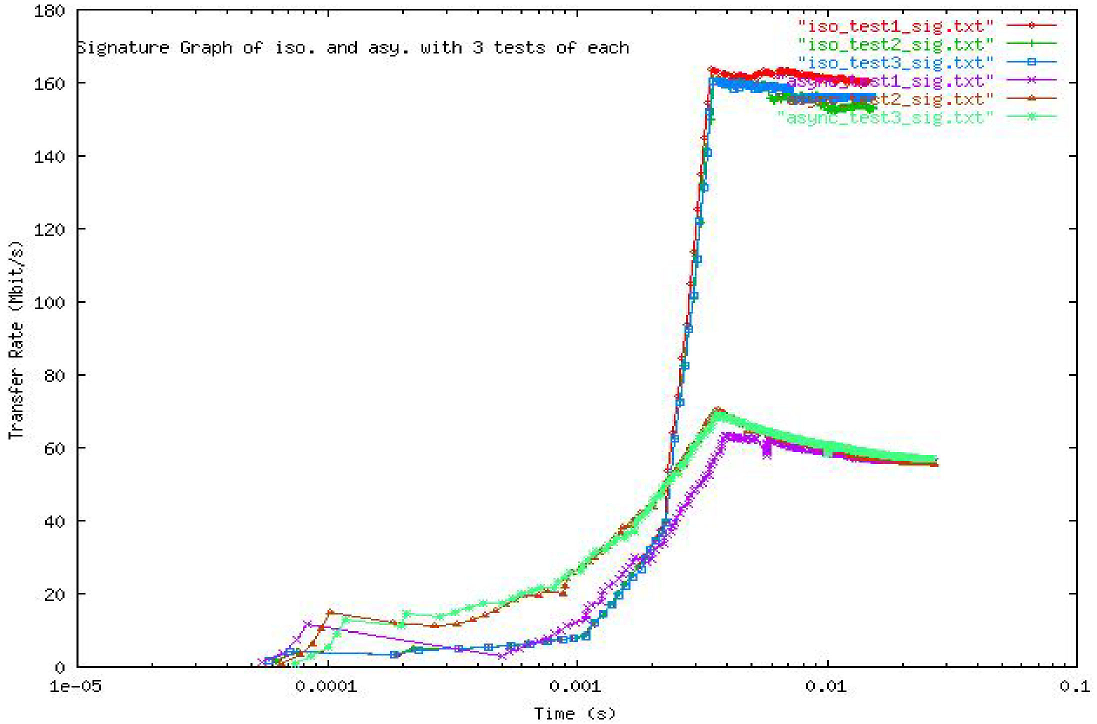


Fig. 5. IEEE 1394 signature over raw driver.

When comparing all the curves, the most interesting in the throughput curve is the comparison with Fast Ethernet (Fig. 6). The maximum transfer rate value for the Ethernet curve is about 80 Mbit/s or 10Mbytes/s.

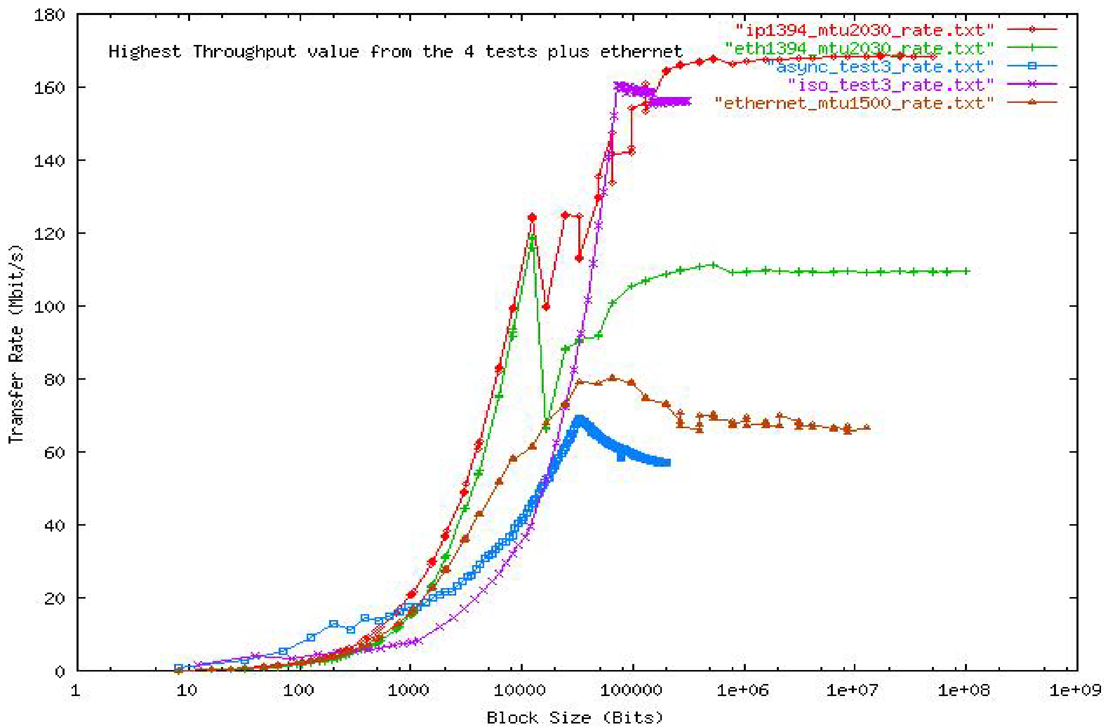


Fig. 6. Comparison of IEEE 1394 throughput for all protocols.

Compared to ip1394 with MTU 2030, which is the fastest IEEE 1394 setup, the Ethernet throughput value is less than a half of it (169 / 80 = 2.1). In other words IEEE 1394 is twice as fast as Fast Ethernet. Fast Ethernet is just slightly faster than asynchronous transfer is. Isochronous transfer over raw1394 is slightly slower than ip1394 but much faster than eth1394.

Tests for Multicasting

Socket Application. The objective is to show the influence on performance of using multicasting compared to unicasting and broadcasting for a socket based application. We also compare IEEE 1394 to Ethernet for the different sending schemes. The solution of a linear set of equations using Jacobi method was used as a benchmark (Pacheco, 1996).

In these tests *multicasting* using *UDP* is unreliable, so is *broadcasting* using *UDP*, and TCP cannot be used for broadcasting. *Unicasting*, on the other hand, is implemented using *TCP* for the purpose of seeing the difference in reliability and speed.

In these tests all three computers were used. One computer hosted the server and the others hosted the clients. All computers were doing the calculations and the server calculated the time to send data for each iteration.

A block size of 8 bits applies to all tests using the Jacobi algorithm. Figure 7 shows the *transmission time* versus the *number of iterations*, for UDP and TCP protocols. In each iteration 800 bits are sent because the matrix size is 10*10 and each number is 8 bits.

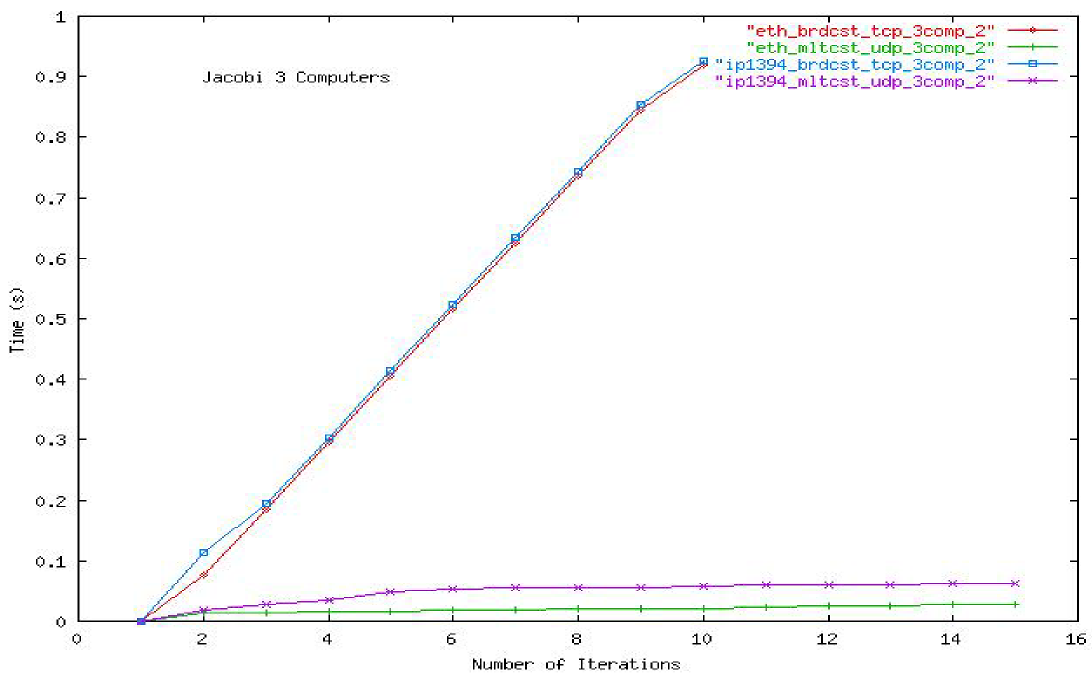


Fig. 7. Socket based Jacobi computations vs. number of iterations.

In each iteration the part of the matrix calculated by the server is sent to the client and the part calculated by the server is sent to the client. This means the whole matrix is sent once. For each send/receive transaction *8 bits* is sent and that is therefore also the *block size*.

When the matrix size, N*N, has been changed from 10*10 to 200*200, data size transferred for each iteration significantly increased. This is reflected in computing transfer time so that each point represents the total number of bits sent to complete the Jacobi algorithm for that matrix size.

Figure 8, where 3 processes are used, shows multicasting faster than broadcasting, and Ethernet slightly faster than IEEE 1394. Using only three computers fails to clearly show this fact and more computers would be needed to prove it. Unfortunately, more computers than three means sending more data via UDP, which is unreliable, and the Jacobi algorithm would probably fail, so it would be virtually impossible to do the tests. To do this kind of test, a reliable protocol or another type of application would be necessary that is not as sensitive to the unreliability of UDP.

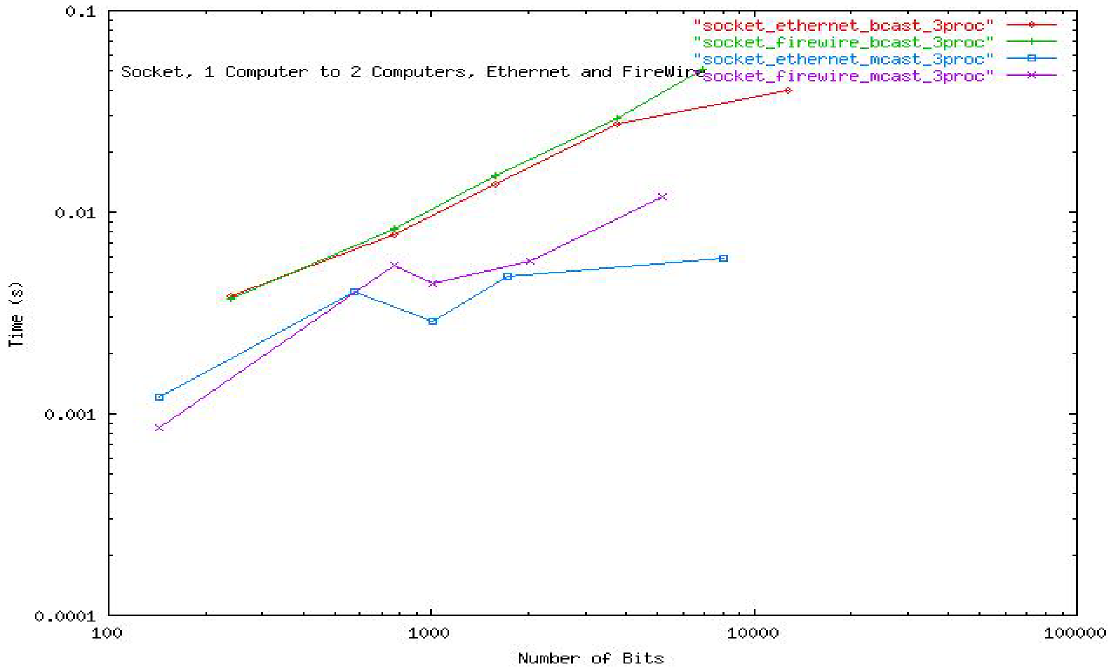


Fig. 8. Broadcasting and multicasting over IEEE 1394 and Ethernet.

MPI Application. The objective is to show the influence on performance of using multicasting compared to unicasting and broadcasting for an MPI application. We also compare IEEE 1394 to Ethernet for the different sending schemes. Results from using a slightly rewritten standard MPICH example are shown (MPICH, 2003).

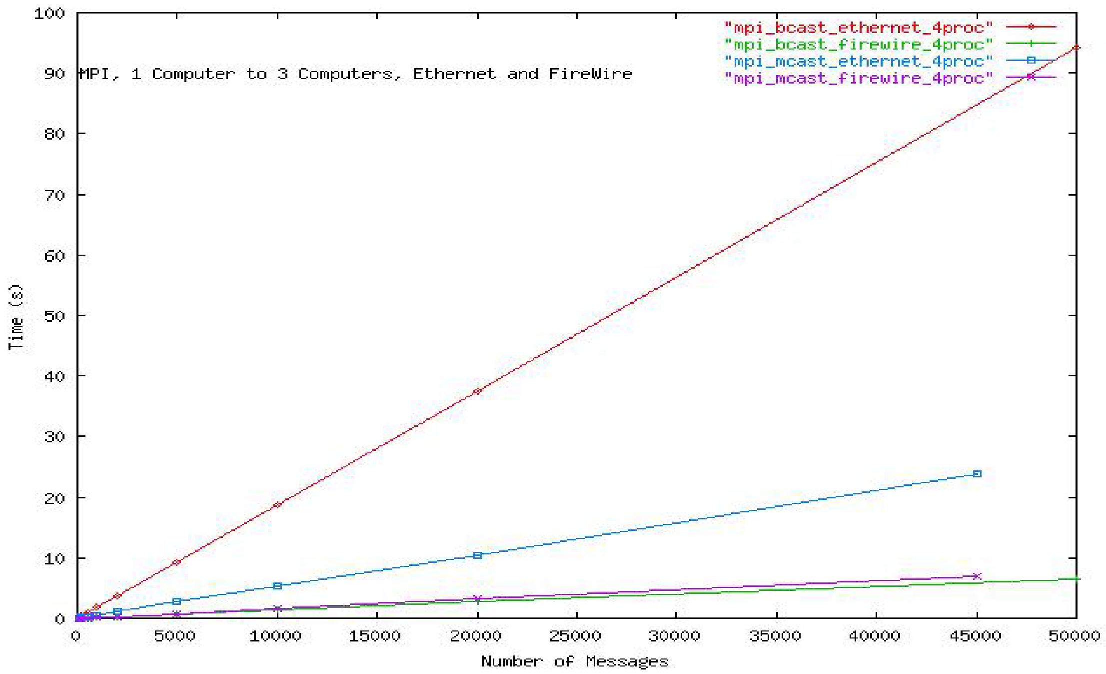


Fig. 9. MPI transmission using broadcasting and multicasting for IEEE 1394 and Ethernet.

This application sends data using the MPI_Bcast() function. It broadcasts and multicasts, respectively, depending on how MPICH environment is set up. Input values are *packet size* and *number of packets*. In all the tests below packet size is kept constant at a value of 1000 bytes and the number of packets are altered. The maximum packet size for the current version of the multicasting setup is 1012 bytes. This is because the UDP packet size maximum is 1024 bytes and the multicasting header is 12 bytes. The maximum numbers of packets for the multicasting setup that can be sent are 49999. This is set so no memory problems will occur. This is, however, not a limit for broadcasting.

Using four processors best shows the difference between multicasting and broadcasting and that multicasting is faster as expected. This is illustrated in Figure 9. In IEEE 1394, multicasting is also closer to broadcasting, but still slightly slower. It is expected that multicasting will get faster than broadcasting with more computers

Figure 10 shows the transfer time versus number of computers, for sending 1000 bytes 20000 times. It shows that multicasting will keep almost the same time, no matter how many computers are used, while on the other hand broadcasting will more or less double the transfer time depending on how fast the computers are. Having a few more computers would show this more clearly, especially for IEEE 1394, but the principle is shown even for 4 processors and Ethernet is showing this pretty clearly. This graph also shows that IEEE 1394 is faster than Ethernet.

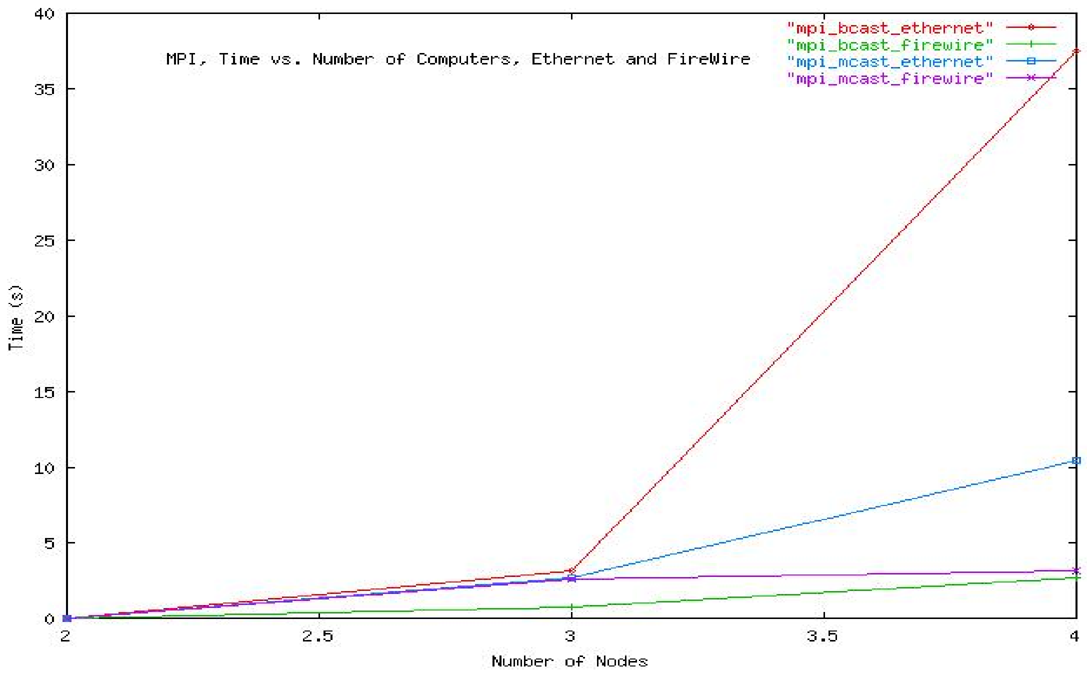


Fig. 10. MPI transmission for increasing number of computers.

Conclusion

This study investigated effects of isochronous transfers and multicasting on cluster computing over IEEE1394 (FireWire). For isochronous mode, the performance of two drivers providing the IP protocol over the IEEE 1394 bus, ip1394 and eth1394, was tested. The results showed that ip1394 is slightly faster than eth1394, when the MTU value was optimized. The speed for ip1394 reached 169Mbit/s or 21.1Mbyte/s.

A third driver, raw1394, was also tested to compare isochronous and asynchronous modes. The tests clearly show the advantage of using isochronous mode over asynchronous mode. The maximum speed reached was 163Mbit/s or 20.4Mbyte/s for isochronous mode and 75Mbit/s or 9.4Mbyte/s for asynchronous.

These results also suggest that there is still a lot of improvement needed for IEEE 1394 software for Linux to reach the maximum theoretical speed of 400Mbit/s. The tests shows that IEEE 1394 is clearly faster than Fast Ethernet, if set up correctly, but the difference might not be enough to upgrade from Fast Ethernet to IEEE 1394 in a functioning network.

The *multicasting* tests performed using *sockets* with the *Jacobi* algorithm showed clearly the unreliability of UDP, as the Jacobi algorithm stopped when a packet failed to go through. The advantage of multicasting, especially compared to unicasting has been demonstrated. The tests for MPI were conducted with multicast over UDP, but with a second layer providing reliability by sending confirmation packages over TCP. The test shows that multicasting is faster than broadcasting and there are no signs of lost packets.

The most obvious extension of the tests, is to add more nodes. This would more clearly show the advantage of using multicasting over broadcasting. In addition, implementing a reliable protocol (such as XTP) for using multicasting over sockets would have a significant impact on validating the results. Finally, improvements in the IEEE 1394 drivers need to be done to achieve speed closer to the theoretical maximum, 400 Mbit/s.

References

Anderson D., (1998). *FireWIre System Architecture – IEEE1394*. Reading, Mass.: Addison-Wesley.

IEEE Std. 1394-1995 High Performance Serial Bus. (1995). IEEE: New York.

MPICH-A Portable Implementation of MPI. (2003). Version 1.2.5.2, Argonne National Lab.: Argonne, Ill. http://www-unix.mcs.anl.gov/mpi/mpich/

NetPIPE; Network Protocol Independent Performance Evaluator. Release 2.4. (1997). Ames Laboratory, Iowa State University: Ames, Iowa.

Pacheco, P.S. (1996). *Parallel Programming with MPI*, Morgan Kaufmann: San Francisco.

Presentation of the initial plans for

the Third Swedish-American Workshop on Modeling and Simulation
SAWMAS-2005
Norrköping, Sweden, August 9th to 11th, 2005

Presented by
Sören Palmgren and Johan Jenvald

Introduction

The Swedish – American Workshop on Modeling and Simulation, SAWMAS-2005 will be the third one in a series of workshops representing the co-operations between a number of people and their institutes and companies in Europe and the USA. The exchanges of ideas and experiences in developing tools and methods for modeling, simulations and analysis with a countless number of applications in the society as a whole is the main driving force for coming together in the name of SAWMAS.

In this presentation we will give foresights in various respects of what will be possible scenarios regarding the third SAWMAS conference to be held. The time and the place of the conference as well as the subjects and the people and their organizations involved will be in focus.

The Time and Place for SAWMAS-2005

SAWMAS has now had the opportunity to be held twice in the beautiful Sunshine state of Florida, the native country of alligators and other exotic creatures. We have all enjoined that. It is now time to move SAWMAS over the Atlantic Ocean to Sweden.

Although the third SAWMAS meeting will take place in the southern part of Sweden we are still in the very northern part of Europe. If you take a look at a globe you will find that the latitude 60° - for example - defines the southern boarder of the Yukon Territory of Canada. We also know that the ocean current of the Gulf Stream makes the difference – so far. The change of the climate due to the warming-up of the earth improves the situation in Sweden even more. Another characteristic of Sweden during summertime is that the sun rises at 4 am and sets around 10 pm. The famous Nordic light can be found in many paintings from different ages.

Essentially, the prospects for a SAWMAS meeting in Sweden from august 9th to august 11th in the year of 2005 could be as interesting and exciting as the first two workshops held in Florida.

The location we have chosen for SAWMAS-2005 is an almost 100 years old former summer residence of prince Carl of Sweden, son of the king Oscar II of Sweden and his wife Sofia.

The main building shown to the left is a spacious manor-like house named Fridhem – meaning ”peaceful home”. 15 years ago a new owner renovated the building and turned it into a comfortable hotel and conference center by adding two annex buildings on the north side of the main house.

The residence is situated on the boarder between a wild and mountainous 1 million acre forest named Kolmården to the north and the entrance to the unique archipelago in the Baltic to the southeast.
The archipelago is called “Blå Kusten” – meaning “the blue coast”, and stretches from a small community named Arkösund down to the town of Västervik, some 60-70 miles with thousands of scattered islands of different shapes and sizes.

If we want to take a half-day off during SAWMAS-2005 – preferably on the afternoon and evening Wednesday 10th– there are some mid-size vessels that can support us with dinner programs and guided tours in the beautiful archipelago.

The water temperature in the archipelago of the Baltic in the beginning of august is often suitable for swimming – about 70 degrees Fahrenheit – but could vary a lot mainly due to wind directions. Wreck diving in the Baltic becomes unique, as the brackish water of the Baltic (about 0.7% salinity) prevents the shipworm Teredo Navalis to survive. Instead the water and the mud preserve the ship. Therefore scuba diving on different wreck sites were you can find wooden ships several hundreds year old, is one of the most exciting adventures you can have with a diving-suite on. However, all such expeditions have to be arranged on an individual basis.

The Subject and the People of SAWMAS-2005

The important part of the next-coming meeting is the further development of the workshop in different aspects.

The basic idea of SAWMAS remains: to gather scientists and kindred working with different aspects of modeling, simulation and analysis and be able to exchange ideas and experiences of good and bad in the process of developing new tools and methods in this field.

However, the ubiquity of modeling, simulation and analysis has – in a very natural way – led to that a large portion of the development of tools and methods are carried out in strong connection with a specific application. A conference like SAWMAS can have a clear role in making these efforts knowledgeable in a somewhat broader sense and possibly widen the area of applications.

Another possibility to deepen the outcome of the SAWMAS workshop series is the possibility of generating cooperation at various levels between the organizations that we belong to. One example is the creation of a meeting place for PhD students in the modeling and simulation area, where they can meet, discuss and present their work, research and findings. The growth and deepening of such co-operations can be initiated in a very natural way through the meetings of SAWMAS.