Information & Security Volume 3, 1999

Modeling and Simulation

Edited by Valentine Penev

Simulation of moving platforms

Status of M&S in Bulgarian MoD

Enabling technologies and education

I&S Monitor

I&S Library Update

I&S Reference Files

M&S Terminology

Acronyms

M&S Related Web sites

I&S Research Centers

Institute of Control and System Research

Author: **Editorial**
Title: **Modeling and Simulation in Defense and Security. NATO Context.**
Year of issuance: **1999**
Issue: **Information & Security. Volume 3, 1999**
Hard copy: **ISSN 1311-1493**

MODELING AND SIMULATION IN DEFENSE AND SECURITY

NATO CONTEXT

Modeling and Simulation (M&S) are not new phenomena in the world of the military. Military training has always involved certain ways of simulating the reality of the battlefield. Depending on the level at which simulation was done it also involved more or less sophisticated modeling activities. The simulations could cover various aspects such as maneuvers, weapons used, own weapons as well as those the opponent was supposed to use and so on. The possibilities of modeling and simulation in the training area became well known after the Great explosion in the computer technology. Now the second stage in Modeling and Simulation is presented and this is the use of the Net for simulation.

Still, a policy for developing and using modeling and simulation all over the Euroatlantic Alliance has not been developed yet. For a number of years nations continued their own policies, in more or less isolated manner, and cooperation was limited to some specific technological domains. As a broad subject, simulation was first introduced on the international scene as a Common European Priority Area (CEPA) within the EUCLID program, then part of the IEPG, later part of the WEU WEAG activities. In the NATO context simulation came up again as a special topic in a symposium on this subject, organized by the US for the NATO DRG in September 1995. Clearly there was coordination, but not in the sense of a focussed effort linking modeling and simulation together.

NATO Involvement

The growing importance of modeling and simulation was recognized by NATO through the establishment of a special Steering Group on Simulation Policy and Applications (SGMS) by the Conference of National Armament Directors in 1996. This group was to carry on earlier work by a special expert group. The SGMS was charged in particular with the task of formulating a NATO policy on this subject, in particular including a Master Plan that would be the guide to get deeper involvement by NATO nations and NATO itself. Of particular importance was the issue that the various national developments should be guided in such a way that interoperability between various modeling and simulation systems would be assured to the maximum extent possible. This seemed all the more important as simulation, as a result of the rapid spread of information technologies, is more and more used on national levels. For the Alliance, the integration of national assets into Allied Command Structures is of prime importance. The national modeling and simulation developments and capabilities should take this into account and make sure that simulation and training activities can be conducted at allied levels.

Given the many technological aspects of M&S and the earlier involvement of DGR groups in this subject, the CNAD decided to have the SGMS report to it through the new Research and Technology Board (RTB) that was officially inaugurated in November 1996. The SGMS did report its progress to the RTB during its meetings and in September 1998 the final report of the SGMS was presented to the RTB at its meeting in Athens.

The SGMS had made tremendous progress in the time allotted to it. The acceptance by the whole SGMS of High Level Architecture (HLA) as the common baseline could be called a small miracle. Proposals for other activities to forward the M&S work are all of good quality. In organizational terms, the very nature of M&S makes it necessary to embed follow-on activity with the military user, the acquisition community, the C3 community and the world of research and technology. It is therefore not surprising that ultimately the CNAD, the Military Committee and finally the North Atlantic Council opted for the attachment of the follow-on work to the Research & Technology Organization (RTO). The NATO Modeling and Simulation Group and the supportive staff of the Modeling and Simulation Co-ordination Office have found their place in the RTO. With the first NATO/PfP Conference on Modeling and Simulation in the fall of 1999 in Norfolk, VA, the RTB really started focusing on the M&S program.

Vision Statement

Modeling and simulation will provide a readily available, flexible and cost-effective means to enhance NATO operations dramatically in the application areas of defense planning, training, exercises, support to operations, research, technology development and armaments acquisition. This goal will be supported by Bulgarian MoD effort that promotes interoperability, reuse and affordability.

Guiding Principles

Based on the vision statement a number of ‘guiding principles’ were formulated. Guiding principles are fundamental, enduring tenets that shape M&S development strategies and implementation decisions to realize the above vision. The keywords in these guiding principles are the following:

- cost-effectiveness: better, faster, cheaper
- consistent with "train as you fight" principle
- means to an end, rather than an end in themselves
- allow the linking of live, virtual and constructive
- interoperable and reusable
- exploit the industrial capabilities of Bulgaria
- verified, validated and accredited
- accommodate security needs of NATO and nations
- foster co-operation, respect for national prerogatives

The NATO M&S Master Plan

In order to guide the efforts for Bulgaria and to give these efforts a structure, a Master Plan for M&S is under development. This Master Plan will contain a number of elements. Apart from guidance to use High Level Architecture type language, it provides for educational options and for a management structure.

The Master Plan will outline NATO's effort to co-ordinate, approve and apply cost-effectively the collective modeling and simulation capabilities and activities with the Alliance. This plan is expected to evolve as the Alliance and its member nations proceed with its implementation, gain experience and develop additional insights.

Simulations, whether pure software or manned simulators, must also be able to interoperate with various real-world, or "live," systems, such as communications and information systems, weapon systems on instrumented ranges, and system components on test benches. This will facilitate the test and evaluation of the live systems and deliver training, course of action, analysis tools, and mission rehearsal capabilities to humans operating those live systems. NATO will need many different simulations, but for cost-effectiveness, it needs the flexibility to re-use them to the maximum extent, building new representations only when existing simulations cannot meet the need. To get the greatest return on investment, NATO must be able to team these representations and interoperate them in different combinations, called federations, to satisfy its diverse and ever-evolving set of user needs.

NATO should lead the planning and integration of such federations, but leave most of the work on simulation development to the nations. We should exploit our industrial capabilities, reusing and modifying commercial and government simulations before we expend resources to build new systems.

Objectives

The M&S Master Plan contains five objectives. The first objective is to establish a common technical framework to promote the interoperability and reuse of models and simulations between the various users of M&S. An important achievement here was the relative early adoption of the 'High Level Architecture' as the standard technical architecture for NATO M&S. This was of particular importance as protracted battles over whose standard is the best would certainly have prevented to achieve the aims set out by NATO. Based upon this choice data interchange standards can be defined.

A second objective of the Master Plan is to provide common services in NATO M&S. This objective encompasses the compilation of M&S information, the provision of key education in the field of M&S, the establishment of a Library and of a Help Desk. All in all this is perhaps an ambitious objective, or rather set of sub-objectives, but a necessary one in order to start some common NATO infrastructure and common use of M&S.

The third objective is to develop simulations. This is typically easier stated than realized. In order to achieve real simulation on a NATO level several sub-objectives are identified. Determining priorities and requirements amongst member states and NATO itself is a first such sub-objective. The identification of strategies (existing ones or under development) in the nations is also important in this perspective. Given strategies, requirements and priorities, the next step is the allocation of resources. As funds are typically limited and as their use is often intended as a facilitator for cooperation between member states, the resources available in the nations themselves are by far the more important ones. Having accomplished these sub-objectives, projects can be set up and brought to execution. Lastly, it is of importance to build in appropriate feedback mechanisms on these projects right from the beginning.

A fourth objective, clearly different from the third one, is the use of simulations. The simulations considered under the third objectives are in a certain sense no more than tools. The usefulness of the tools is proven by their application. What has been mentioned on the objective of simulation development is to a large extent also true for the exploitation of simulations. Again planning, prioritizing, the allocation of resources, are important factors. Important also is that those who intend to use these simulation tools provide the necessary databases. Again, after the actual running of the simulation, feedback is essential: has the simulation provided some of the answers looked for, did it provide added value, and so on.

The fifth objective addresses that we live in a dynamic, and not a static world. It is important to monitor and, if appropriate, incorporate technological advances as they become available. Certainly in this field, rich in Information Technology, rapid turnovers of some of the constituting technologies can be expected. Technology advances will provide opportunities to increase functional capabilities, performance and overall M&S effectiveness.

Summary of assessment

The Steering Group conducted a baseline assessment of the status of M&S within the Alliance, which provided a high-level overview as of 1997. It is from this baseline that the Alliance moves forward to reach its M&S vision. The assessment was: "NATO has not yet begun to harness the full potential of M&S to improve operations. Uses are rich and diverse, but a common Alliance strategy incorporating interoperability and reuse, a key to cost-effectiveness, does not exist yet."

This earlier assessment is still up to date. While start of the M&S work has been slower than originally intended, it has now become clear that Modeling and Simulation will play an important role in the Alliance’s Defense Capability Initiative. Also, M&S is going to be an important tool in the activities of the NATO Training Group, in particular for the Training and Education Enhancement Program. Computer Aided Exercises will further become more and more common.

Working together as military users, logistic support organizations and technologists will help Bulgaria to make necessary progress in the preparation for NATO integration. This way in the near future Bulgaria will posses increased capability for compatible and more effective planning and operation of its forces.

In the focus of I&S

In an attempt to cover the broad area of Modeling and Simulation in Defense, we chose the set of articles in the current volume of I&S.

The first article enables the readers to learn about military simulations and how they are used not only in defense, but also in crisis management in international context. It further updates the reader on the latest developments in the military simulation and training industry. The focus is on the command laboratory simulation but defined in the context of crisis management center concepts.

The next paper describes other interoperability challenges that federation designers and managers face. These challenges relate to the differences between modeling techniques employed by different simulation federates and the logical interpretation of data shared among federates. Using the Federation Development and Execution Process as a guide, the authors identify the key areas where federation developers address simulation interoperability as they apply High Level Architecture to their domains particularly in battle space simulation applications.

The third paper shows the benefit of using COTS software in the RENAISSANCE efforts, which are construction of reusable, generic, software "building blocks" based on legacy software. In theory, new missions could then select among these building blocks to build systems in an object-oriented manner. The focus is on applications in designing space mission packages.

The first three articles provide examples for the use of simulation on grand strategic and strategic levels. The fourth article in the volume provides an excellent ‘tactical’ example. It describes distributed simulation as an approach to building large-scale simulation models from a set of independent simulator nodes communicating via a network. The U.S. Army uses distributed simulation systems for both training and analysis. Those systems include both crewed simulators and computer generated forces (CGF) systems; the latter use software, rather than human crews, to generate the behavior of entities in the simulated battlefield. CGF systems must include algorithms for all of the tactical behaviors that are needed for the simulation. One such tactical behavior is "Fire Zone Defense." An algorithm for this behavior must select defensive deployment locations on the terrain for the individual entities (e.g., tanks) of a unit (e.g., a company) to effectively defend an assigned engagement area. The entities of the unit then move to those locations. The paper presents a new algorithm for the behavior combining a geometric terrain analysis algorithm with a greedy optimization algorithm.

The second group of papers presents views and activities of Bulgarian scientists in the area of simulation of moving platforms. The articles examine the main steps in the process of creating a simulator for moving platforms. This principal task includes the following items: understanding the behavior of the simulated object, appropriate mathematical modeling and 3-dimensial visualization. The general features of a fuzzy controlled semi-automatic missile are studied and missile response against the mobile target is evaluated and discussed. The main characteristics of simulator for antitank wire guided missile designed with DirectX7 SDK are presented.

The third group of articles presents the status of modeling and simulation in the Ministry of Defense of Bulgaria and a sample of R&D activities of Bulgarian military and civilian specialists. This group is intended to underline the necessity of conceptualizing and implementing NATO compatible Bulgarian policy towards M&S for defense purposes. First, we need to start a discussion on the national problems in Modeling and Simulation. This discussion is expected to involve Bulgarian and NATO experts and to proceed with initial concept implementation, to gain experience and develop additional insight. It is essential that all M&S experts should devote some time and thought to the consideration of the major problems with which we are currently faced. Some of them can be solved with relative rapidity given common sense and a correctly appreciated self-interest. Others will require foresighted planning and patience as, one by one, the necessary steps are taken, leading to readjustment of concepts and inauguration of new attitudes to Modeling and Simulation. The key to current troubles stems not only from the economic difficulties and limited budgets in the last ten years, but more so from the rigid thinking, from the culture of taking without giving, accepting and not sharing, grasping and not disseminating. The goals of the concept are to clarify the real state of Defense Modeling and Simulation in Bulgaria and to establish the initial conditions for creating a Modeling and Simulation Master Plan. Meeting the military requirements will be an evolutionary process as needs are further refined across the spectrum of mission areas. M&S support will concentrate on activities at the operational and simulator levels. We think that we have one main goal; fortunately or not, we also have many ways to reach this main goal.

The last group of papers is devoted to enabling technologies and education in M&S design. In the first article, an approach to the next generation televirtual environments that integrate collaboration with distributed computing and modern modeling and simulation technologies is presented. The authors follows the 3-tier architecture with the Web Object (Java/CORBA) based middleware, VRML/Java3D/DirectX based front-ends and JDBC/PSS/OLEDB based back-ends. The design is tested and the integration concepts by prototyping a multi-user authoring and runtime environment to support WebHLA based distributed military simulations. The authors present taxonomy of collaboratory frameworks and integration paradigm, based on the WebFlow system.

In the next paper an idea for distributed designer for automatic control systems is presented. This investigation may be implemented to develop simulators of moving platforms in the group work.

The last paper illustrates some of the interesting concepts that are explored in the course "Military Simulation Techniques & Technology". This is a 3-day training course designed to teach engineers and project leaders the essential techniques required to design, build, and operate a military simulation system.

As usual, for readers, interested to learn more about foal topic of the I&S, this volume presents five books, as well as a significant number of Internet addresses.

BACK TO TOP

Information & Security. An International Journal
e-mail: infosec@mbox.digsys.bg

Author: **Roland J. Roland**
Title: **Applying Modeling and Simulation to Enhance National and Multi-National Cooperation**
Year of issuance: **1999**
Issue: **Information & Security. Volume 3, 1999**
Hard copy: **ISSN 1311-1493**

APPLYING MODELING AND SIMULATION TO ENHANCE NATIONAL AND MULTI-NATIONAL COOPERATION

Ronald J. ROLAND

Table Of Contents:

Introduction

This paper is the first in a series that will describe the application of modeling and simulation to enhance country-to-country, agency-to-agency and coalition-to-coalition cooperation and understanding. The goal of the series is to provide information on the availability of M&S applications, their potential benefits and associated cost, develop a hypothetical plan for the implementation and use of various M&S applications and, finally, encourage a dialogue of understanding about the benefits and requirements to obtain, implement and execute M&S tools.

What really generates success in civil/military operations? Many cite technological superiority in weaponry, intelligence of the adversary's intentions or sometimes just raw numbers. Yet the greatest victories and successes in history have usually involved smaller forces defeating larger ones, frequently against odds that no gambler would take. What is the critical edge that allows smaller forces to defeat their more powerful opponents? Part of it is experience, though this is frequently bought at the price of lives and lost opportunities. The truth is that in lieu of tempering units in the forge of battle, the crucible of training is what can make the difference in combat and in civil operations. Training is frequently viewed by some leaders, military and non-military, as a wasteful misuse of funds that might be better used to sustain larger forces or to buy new and more effective equipment. More thoughtful individuals realize, though that often the quality of the operator can overcome the capabilities of better equipment in the hands of poor operators. This is the training philosophy that has driven nations like the United States and the United Kingdom for the last generation, and has allowed so many victories and successes.

Herein we provide a background concerning one effort to proliferate a common architecture for crisis management, a history of M&S development and an introduction to a specific software application that is widely used to train senior staffs within simulated crises situations. The focus shall be on computer aided exercises, training and analysis. It will evolve through a series of articles. The intent is to convince the audience that the use of M&S will provide them an efficient and effective means to maintain their edge to better manage or perhaps avoid future conflicts.

A National Crisis Management Command Center

If you think education is expensive, try ignorance.

Unknown teacher

Although the demise of the Warsaw Pact in 1989 has changed the political landscape of the world, natural disasters and other crisis situations continue as hostile threats to all nations. Devastating earthquakes, floods, environmental disasters and a host of other catastrophes may cause devastation with loss of lives and considerable property damage. Illegal activities such as drug trafficking and smuggling also occur at all levels. An important issue is that none of these events recognize political boundaries. In response to these situations, departments, agencies and nations need to effectively apply local, regional, and national resources to manage the consequences of disasters and crises. Increasing in importance is the ability to work across national boundaries to collaborate and inter operate with multi national resources. These resources typically include some combination of civil and military units that may be called upon to provide assistance in the face of crisis situations.[1]

In order for these resources, both civil and military components, to respond to crisis situations in an efficient manner there are at least three fundamental requirements; (1) the availability of information regarding crisis situations and military/civil resources readiness; (2) coordination among the organizations and agencies (intra and international) involved in crisis management; and (3) continued training and exercising of the resources so that they can respond effectively when needed. Herein we address the operational aspects of an information support system intended to assist national Ministries of Defense (MODs) in coordinating with other national (and regional) organizations dealing with crisis situations and also in applying military and civil resources in execution of crisis management responsibilities in coordination with other national and international organizations.

In the course of executing Command, Control, Communications and Computers (C4) studies for Central and Eastern European nations, a common thread emerged from the analysis of national C4 system requirements and on-going modernization plans. All nations involved in the studies were engaged in planning for the introduction of centralized information collection and processing systems to support the management of military forces in crisis situations. Because these systems were being planned independently, there was little commonality of system concepts or system architectures. Consequently, the ability to share information within and among nations in a regional crisis and to collaborate in crisis relief actions would, most likely, be severely limited.

In response to the apparent need for a centralized crisis management capability and in the spirit of the Regional Airspace Initiative, which resulted in an Air Sovereignty Operations Center (ASOC) program, the U.S. Air Force Electronic Systems Center (ESC) developed a concept for implementation of a national command center for crisis management. This command center, identified as the NMCC, would support both national civil and military crisis situations and, because different national systems would be built on a common architectural platform, it would also support regional collaboration in response to regional crisis situations. This NMCC concept was presented to several nations in the spring of 1999 and met with favorable response. The presentation also included a discussion concerning a software component, the Joint Theater Level Simulation (JTLS) to be considered as a baseline for training and analysis at the national and multi-national command or decision making levels.[2]

Based on the initial favorable response, the U.S. government formally introduced the new policy initiative to Partnership for Peace nations at a multinational conference in Sofia, Bulgaria in June 1999. As described in the U.S. keynote address at the conference, the NMCC is intended to provide national command authorities with a modern, integrated command and control center to support decision making in the event of civil or military crises. Further, the NMCC will be built on a NATO-compatible technical architecture platform and will provide interfaces that are compatible with comparable NATO and U.S. command and control systems.

Seven countries at the conference indicated they would collaborate in the initiative, most as active participants. Since that time, several other countries have expressed interest in the initiative. The next step in the process of advancing the initiative—forming a multinational Working Group of potential program participants—occurred in Predeal, Romania in September 1999. This Working Group will modify and agree this Concept of Operations (CONOPS) for the NMCC and also establish consensus on a technical architecture framework for the NMCC. These two documents will serve as the foundation for an acquisition effort for those nations choosing to participate in the NMCC program.

Modeling and Simulation Support

The NMCC Concept of Operations (CONOPS) does not include a conceptual model for modeling and simulation (M&S). As we know, M&S has been extensively and successfully applied to a wide range of civil/military problems including combat, operations other than war, acquisition, decision making, contingency plan analysis, logistics and communications.[3] I has been used as a decision support tool to evaluate how a civil/military force should be constituted, how it might be deployed, and how its organic resources or weapon systems should be acquired and maintained. The crisis management regime often requires domain experts because the development and use of simulation models to support an NMCC require specialized knowledge in unique problems. Most military models have been developed by highly specialized groups and used in narrowly focused user communities. For example, the Army is interested in wargaming simulation with ground forces. The Navy is interested in battle group simulation with aircraft carriers, aircraft and ships. The Marine Corps' interest is in amphibious operations, and the U.S. Air Force is working on space systems, strategic, long-range bombing and tactical air-to-air and air-to-ground support. Each service adopts different logistics systems for weapons systems maintenance; the Navy and the Marine Corps follow three levels (organizational-level, intermediate-level, and depot-level) of maintenance, while the Air Force uses two levels of maintenance, and the Army has five levels.[4]

As a result, there are many organizations and agencies involved in M&S. Each community uses its own special jargon, abbreviations, and acronyms, which makes it difficult for the various government agencies and services to communicate. More than 150 pages of *Glossary of DoD M&S Terms* are available from Department of Defense (DoD) Directive 5000.59-M.[5] Although the simulation community is huge, there is a lack of systematic communication and no central resource library. Many models have been developed on a stand-alone, system specific, as-needed, and as-afforded basis, which has resulted in redundant investments. Typically, more efforts are spent to develop a new simulation system and building the simulation infrastructure than the efforts to develop components specific to purpose of the simulation. If the infrastructure and other simulation components could be reused, the payoff would be enormous. For this reason, object-oriented programming and the high level architecture (HLA) are getting more attention.

Although M&S has been used to investigate military problems for many years, there is very little literature available for general readers who do not specialize in military simulation. Military simulation models are different from others because (1) many of them are highly classified with details that could not be widely disseminated; (2) weapon capabilities and use are not typically used in other M&S; (3) certain algorithms are closely controlled to avoid reverse engineering by potential adversaries; and (4) the use of certain equations, e.g., Lanchester, which is often used in wargaming simulation, is not typical of commercial M&S. The purpose of this paper is to provide the reader an overview of military simulation and insight to its future directions. We review recent developments in military modeling, particularly in wargaming simulation. We provide sources of many DoD documents including internet homepage addresses, where applicable, so that the reader can retrieve the updated information. Since more and more DoD documents are available electronically via the internet, it becomes easier for simulationists to access information on simulation topis. There are tremendous opportunities for expansion of simulation applications in every aspect of life.

Organizations

The DoD and the Joint Staff maintain their own agencies for M&S. In addition, each Service maintains M&S offices. In June 1991, the Defense Modeling and Simulation Office (DMSO) was established by the Under Secretary of Defense for Acquisition and Technology. Two responsibilities of DMSO are to publish DoD M&S policy and promote cooperation among DoD agencies. In January 1994, the Deputy Secretary of Defense promulgated *DoD Directive 5000.59* to the DoD community. This directive is entitled *DoD Modeling and Simulation (M&S) Management*, and was a DOD-wide effort to establish policy for M&S. It was a significant step toward centralizing the management of DoD M&S activities. In accordance with the DoD Modeling and Simulation Master Plan (DoD 5000.59-P, dated October 1995), DMSO is leading a DoD-wide effort to establish a common technical framework to facilitate the interoperability of all types of models and simulations among themselves and with command, control, communication, computer, and intelligence (C4I) systems, as well as to facilitate the reuse of M&S components. This Common Technical Framework includes the High Level Architecture (HLA), which represents one of the highest priority efforts within the DoD modeling and simulation community.

The DoD Modeling and Simulation Master Plan initial definition of the M&S HLA was accomplished under the sponsorship of the Defense Advanced Research Projects Agency (DARPA) Advanced Distributed Simulation (ADS) program. It was transferred to DMSO in March 1995 for further development by the DoD-wide Architecture Management Group (AMG). Central to this task was the development of a set of prototypes that addressed critical issues in the HLA. In September 1996, the Under Secretary of Defense for Acquisition and Technology (USD(A&T)) approved HLA as the standard technical architecture for all DoD simulations and required all computer simulations for military operations meet the HLA standardization requirements by FY2001. Dr. Kaminski's directive mandated that all DoD simulations, failing to comply with HLA standards by a specified date, be retired from service.[6]

The Executive Council for Modeling and Simulation (EXCIMS)[7] is a high level advisory group on DoD M&S policy, initiatives, standards, and investments. More details about DoD simulation activities and *DoD Directive 5000.59* can be obtained on the DMSO internet homepage.

As was mentioned, each Service maintains its own M&S activities. The Army has long M&S history and is better organized than the rest of the services. Deputy Undersecretary of the Army for Operations Research (DUSAOR) oversees all Army Modeling and Simulation. The Army Modeling and Simulation Office (AMSO)[8] is the operational activity for Army M&S. The Army maintains the modeling and simulation homepage for the Army Modeling and Simulation Resource Repository (MSRR).[9] The Army's National Simulation Center[10] located in Ft. Leavenworth, Kansas, supports simulation training exercises around the world. The Army currently maintains six major simulation models for training. They are Janus, VICTORS (Variable Intensity Computerized Training System), BBS (Brigade/Battalion Battle Simulation), CBS (Corps Battle Simulation), TACSIM (Tactical Simulation) and CSSTSS (Combat Service Support Training Simulation System). The details of these models and a list of other Army simulation models are available in **MO**dels & **S**imulations: **A**rmy **I**ntegrated **C**atalog (MOSAIC).[11]

The Air Force also has a long history of M&S applications. The Directorate for Modeling and Simulation (see http://xoc.hq.af.mil) is the single point of contact in the Air Force for policy on modeling, simulation and analysis activities. It includes the Evaluation Support Division, Technical Support Division, Warfighting Support Division, and Air Force Studies and Analysis Agency.

The Navy and the Marine Corps have smaller M&S organizations compared to the Army and the Air Force. They have a Modeling and Simulation Advisory Council that guides the development of policy, coordination and technical support and promotes the use of the Navy-wide common support services. The Navy and the Marine Corps maintain their own Modeling and Simulation Management Offices (see http://mcmsmo.usmc.mil), and set their own M&S policies. A simulation model, the Research, Evaluation, and Systems Analysis (RESA) was developed as a naval warfare command, control and communication (C3) analysis tool for the Navy. The Marine Air Ground Task Force Tactical Warfare Simulation (MTWS) is one of the Marine Corps' tactical combat simulation models.[12]

Classification of Military Simulation Models

According to the Defense Science Board, military simulations are classified into three categories: *live*, *virtual*, and *constructive*. While there is no clear-cut distinction among these categories, it is still helpful to understand the basic differences. *Live* simulation involves real people and real systems. Operational test and evaluation (OT&E), and military field exercises are examples. Live simulations in support of training are conducted at the Army National Training Center (NTC)[13] located in Ft. Irwin, California; the Navy "Strike University" in Fallon Naval Air Station, Nevada; the Air Force Red Flag Site at Nellis Air Force Base, Nevada; and the Marine Corps Air-Ground Combat Center in Twenty Nine Palms, California.

The NTC provides an example of a live simulation. It is a vast expanse of desert approximately the size of the State of Rhode Island. This is where the Army conducts training exercises in order to prepare itself for war in the desert. There are approximately 2,500 soldiers permanently stationed at NTC who function as the "home team." This group pretends to be the enemy and uses all of the doctrine and tactics of the opposing force. The visiting teams arrive at NTC twelve times a year and conduct wargame type simulations against the home team. Every move and every shot fired is monitored by a powerful laser engagement system that records all of the signals from the pieces of armor and other equipment that are participating in the exercise. All of this information is fed into the computer simulation, and numerous statistics are tallied so that, at the end of the exercise, both teams can be evaluated and areas of improvement can be identified.

Virtual simulation involves real people in a simulated system. This includes aircraft and tank simulators.[14] This type of simulation is helpful in training, and for evaluating control, decision and communications skills. Virtual simulation has become more popular with developments in computer technology, especially computer graphics. The journal *Military Simulation & Training*[15] is a good source for up-to-date information on military training simulators. In *constructive* simulations, humans may (or may not) interact with the model and everything is simulated. Constructive simulations of combat include wargames for training as well as for analytical tools. Constructive simulations may be used for training events that range from senior staffs to the operator-level. For example, JTLS is a constructive simulation, which can be used for staff training as well as for operations planning analysis. We now describe JTLS in more detail.

The Joint Theater Level Simulation

The simulation of combat, or a wargame, is used more and more extensively to reduce cost and maintain a trained force. It is an inexpensive alternative to live training exercises. Simulations are also very useful for testing and evaluating proposed procedures, strategies and various systems such as economic, weapons communications and civil architectures. We present and discuss the Joint Theater Level Simulation (JTLS) as an illustration of a wargaming simulation.[16] The purpose is to illustrate various aspects of wargaming simulation using JTLS as an example. JTLS is a theater level simulation that models ground, air and water based resources.

The development of JTLS began in 1983 as a project funded by three Army organizations: the U.S. Readiness Command, the U.S. Army Concepts Analysis Agency, and the U.S. Army War College.[17] It has had continuous functional and system upgrades since that time. Its primary focus is on conventional joint and combined operations and is currently managed by the U.S. Joint Forces Command/ Joint Warfighting Center, Suffolk, Virginia.

JTLS was designed as a theater-level model for commanders and planners as an Operations Plan (OPLAN) analysis tool, support material for education, command post exercise support for training, and a primary means to investigate the results of combat and civil affairs. It is heavily used as an exercise driver where JTLS provides the environment for the dynamic interactions of intelligence, air, logistics, naval, and ground forces. This environment allows users to develop insight into the relative merits of alternative courses of action, force structures, combat systems, and procedures.

The model is currently in use by numerous agencies including the Joint Warfighting Center, the Warrior Preparation Center, NATO’s Command Control and Consultancy Agency (NC3A), the National Defense University, the Army War College, the Naval Postgraduate School, Combined Forces Command Korea, the Australian Defense Force Warfare Centre and the South Korean Institute for Defense Analysis. It is installed at Hellenic National Defense College’s M&S Center, the Defence Evaluation and Research Agency, United Kingdom and the Turkish War College. The Louisiana State University, MITRE Corporation and RAND Corporation have evaluating JTLS for application to non-combatant environments or potential research purposes.[18]

JTLS is a multi-sided, interactive, computer-driven simulation. In this context, *multi-sided* really means that there can be up to ten sides depicting various organizations whether friendly, neutral, hostile, unknown and/or civilian. This is a dymanic environmental variable that is set at each instantiation of JTLS.[19] One recent JTLS scenario includes the sides called the Gulf Coalition, United Nations (UN) Forces, Israel, Iraq and Iran. Each side in turn consists of one of more factions limited by the hardware, scenario requirements and users’ imaginations. The Gulf Coalition factions included Saudi Arabia, Kuwait and one that represents the "civilian populace." All of the UN members were included as factions within the UN side. Factions are also included within the other sides, which permitted an accurate depiction of the forces and perturbational influences within the Gulf region.

Each JTLS side can be subdivided into an unlimited number of factions. A faction’s side allegiance is dynamically changeable during the game (scenario). Side relationship is asymmetric and can also be changed during the game. A large number of players (for example 300) can be involved simultaneously in a single game. The actual number is defined by the User when planning for the training event. Since JTLS can be highly distributed depending on the communications available, the Players can literally be anywhere in the world or all in the same room. In the case of using JTLS as an analysis tool, a meaningful analysis can be managed by as few as two people. JTLS models coalition air, land, sea, amphibious, and special forces operations. The model can support limited nuclear and chemical effects, low intensity conflict, and pre-conflict operations. The model also supports the representation of civilian and non-combatant forces.

The JTLS system consists of six major software modules and numerous smaller support programs that work together to prepare the scenario, run the game, and analyze the results. Designed as a tool for use in the development and analysis of operation plans, the model is theater-independent, that is the data for a specific scenario are stored in a database separate from the source or object code. The database may contain highly classified or sensitive and is purposely maintained independent of the software until execution. The JTLS program itself is unclassified.

Model features include Lanchester attrition algorithms, detailed logistics modeling, and explicit air, ground, and naval force movement. In addition to the model itself, the JTLS system includes software designed to aid in scenario database preparation and verification; entering game orders; and obtaining scenario situational information from graphical map displays, messages, and status displays. The movement of forces within any combat environment is affected by the terrain. The terrain is represented as a hexagonal grid overlay on a map projection. The maximum geographic region or area used in a JTLS scenario is 2,000 by 2,000 nautical miles. The hexagonal overlay design is used to provide an efficient means to calculate and model force movement and to describe both terrain and man-made obstacles. Each hexagon, in the database, is described in terms of its relative geographic location, the terrain within the hexagon boundaries, the elevation, and the barriers on each of the six sides. Hexagon size and the number of hexagons represented in a terrain database are user-data entries. Locations of objects in the game can be displayed as a hexagonal reference, latitude/longitude, or a military grid reference. Objects can be located anywhere on the game surface and are not limited to the center of the hexagons.

JTLS does not require programming knowledge to use it effectively. As an interactive model, it requires human decisions to manage the processes and entities. The players receive messages and reports concerning the movement, attrition, and logistics status of their own forces, as well as intelligence summaries and capabilities of opposing forces. The player at each workstation can elect to view messages in plain language or a special military format. Messages may be electronically sent to standard Simple Message Text Protocol (SMTP) electronic mail workstations. Electronic feeds to several military command and control systems, such as the Global Command Control System, Joint Operational Tracking System, and Joint Military Command Information System, have been demonstrated.

The players interact with JTLS and receive graphical feedback through the Graphics Input Aggregate Control (GIAC). They receive messages through the Message Processor Program (MPP), and status board information is presented by the Information Management Terminal (IMT). These programs obtain their data and communicate with the main simulation component, the Combat Events Program, through software modules called the G Data System, using its data server program, GENIS. A single GENIS (the primary GENIS) is connected to the Combat Events Program using the TCP/IP network protocol. A GENIS may have other GENISes or interface programs as clients. The number of clients that a single GENIS can have at one time is determined by a system parameter of the machine on which it is executing. The parameter defaults to 64 on most machines, and can be modified by system maintenance personnel. A typical player's workstation has a GIAC, MPP and IMT, all operating and connected to a GENIS.

JTLS can be operated on a single workstation, or multiple workstations, and distributed on either a Local Area Network (LAN) or a Wide Area Network (WAN), thus providing a distributed exercise/gaming environment. The computer system support requirements for conducting simulations or analytic excursions using the JTLS model are dependent on the specifics of the event. The purpose of one event can be quite different from another (e.g., analysis, education, contingency plan development, etc.), and could require different support systems. The computing system is a composite of resources such as hardware devices, system software and utilities, communication lines, language compilers and databases.

The JTLS system can be run on a workstation of very limited processing power. For very small test databases, the CEP, GENIS, and two player suites (controller and one side) can be run on a single workstation of the SPARC station 2 class, but system performance is marginal. For exercise applications, in general, each active player requires a workstation of at least SPARC station 5 capability with 32 megabytes (MB) of random access memory (RAM) to perform adequately. For medium-size databases, the CEP and the primary GENIS each should have a SPARC station 20 level workstation with 128 MB of RAM, and each subordinate GENIS should have a workstation of at least SPARC station 20 level processing power, with 64 MB of RAM.

The JTLS source, object, and executable files occupy approximately 550 MB of disk storage. A medium to large database might require another 50 MB of storage. Each checkpoint will use between two and four times as much storage as the initial database, depending largely on the intensity with which player's messages are managed. A 1.3-gigabyte disk devoted to the game directory (with tape backup) is a reasonable starting requirement.

Most of the JTLS system is written in the SIMSCRIPT II.5 programming language. It continuously improves with new technologies. Los Alamos National Laboratories has developed a graphics user-interface. ROLANDS & ASSOCIATES Corporation (R&A)[20] and NC3A have created several tools for the development of scenarios for JTLS. JTLS has been successfully used in conjunction with live training during exercises. For example, the KEEN EDGE (U.S. – Japanese) and COBRA GOLD (U. S. – Thailand) annual exercises both included live training. KEEN EDGE 95 was held at Camp Ojojihara, Japan to introduce U.S. and Japanese Ground Self-Defense Force armed forces and civilian counterparts to each other's way of doing business. COBRA GOLD is a joint exercise with the U.S. and Thailand Forces held in Thailand.

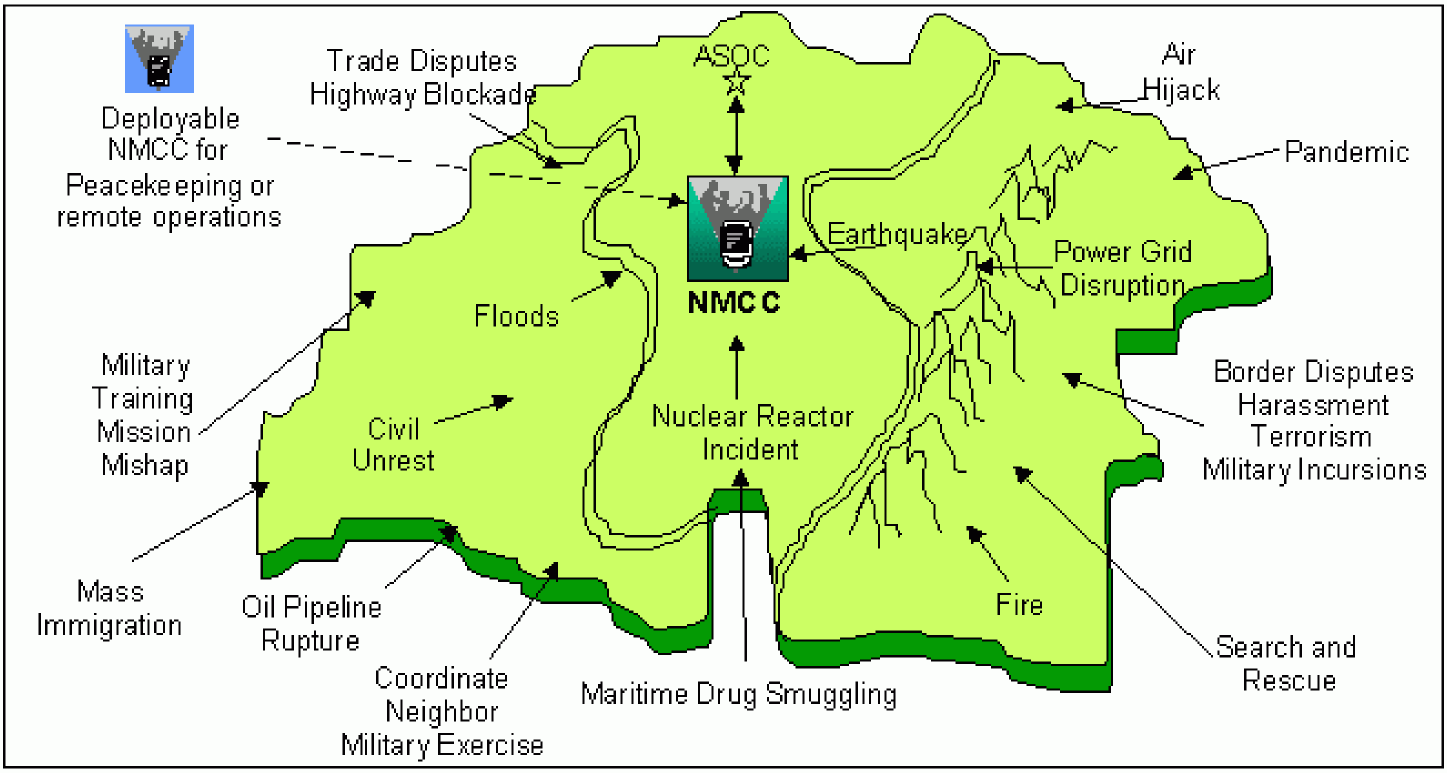


Figure 1. NMCC and JTLS Support to Crisis Management Operations

Future Topics

The NMCC is visualized as a centralized facility to provide national-level coordinated management for military and civil crisis response. It will be initially controlled and operated by the MoD, with civil agency participation/liaisons. The NMCC concept includes interfaces with service headquarters, national military information sources, national civilian agencies/organizations, and regional or foreign agencies/organizations to receive and disseminate information. Figure 1 shows different scenarios in which the NMCC and JTLS might provide support to model and/or manage crisis response operations.

The next piece in this series will focus on the synergism between the potential capabilities of the NMCC and the inherent features contained in the current release of the Joint Theater Level Simulation (JTLS) software. Future topics planned for this series will include discussions on security issues relative to this concept, the future of JTLS within the U.S. M&S community, NATO M&S Policy relative to JTLS and current uses of the JTLS software in various colleges and universities.

References:

1. National Military Command Center Concept of Operations, 22 November 1999 (draft).

2. The Joint Theater Level Simulation, presented by Dr. R. J. Roland and demonstrated by Mr. R. Kalinyak, *International C4/NMCC Conference*, 21-23 June 1999, Sofia, Bulgaria.

3. Keebom Kang and Ronald J. Roland, "Chapter 19. Military Simulation," in *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*, ed. Jerry Banks (Co-published by Engineering & Management Press and John Wiley & Sons, Inc., 1998), 645-658.

4. Please note that in the context of this paper, all references to military or government organizations will be U.S. unless otherwise noted.

5. It can be downloaded from http://www.dmso.mil/docslib/mspolicy/glossary/glossary1195.pdf.

6. Additional details about HLA can be found at http://hla.dmso.mil.

7. See also http://www.dmso.mil/wrkgrps/excims/charter.txt.

8. See http://www.amso.army.mil.

9. See http://www.amso.army.mil/amso2/armymsrr/

10. See http://www-nsc.army.mil

11. See http://hp01.arc.iquest.com/mosaic/mosaic.html

12. For more information regarding the organizations and groups involved in military M&S activities see http://www.dmso.mil/orgs.html#OEGS.

13. See http://www-nsc.army.mil/

14. For an example, see the internet homepage http://www.bgm.link.com/mfs.html for different military flight simulators.

15. Published by Monch Publishing Group, Federal Republic of Germany, ISSN 0937-6348.

16. See http://www.rolands.com/jtls.html.

17. A History of the JTLS Development is available from ROLANDS & ASSOCIATES Corporation.

18. Kang and Roland, "Military Simulation," discusses JTLS for military operations other than war.

19. Kevin Brandt and Ellen Roland, "Modeling Coalition Warfare: A Multi-Sided Simulation Design," in *WSC 93 PROCEEDINGS: WINTER SIMULATION CONFERENCE PROCEEDINGS 1993*, ed. Gerald W. Evans, Mansooreh Mollaghasemi, Edward C. Russell and William E. Biles (Winter Simulation Board of Directors, 1993. Library of Congress Number 87-654182), 977-983.

20. See http://www.rolands.com.

RONALD J. ROLAND, Ph.D., Management, University of Nebraska, Lincoln, NE, 1980 (dissertation on *Using Artificial Intelligence Techniques to Design and Evaluate Decision Support Systems*); M.S., Computer Science, University of Hawaii, Honolulu, HI, 1971; B.S., Mathematics, Colorado State University, Fort Collins, CO, 1969. Dr. Roland is the co-founder and President ROLANDS & ASSOCIATES CORPORATION (R&A). He has managed the development of several computer simulations including the Joint Theater Level Simulation (JTLS), Planning Alternatives for Interdicting National Terrorism (PAINT), and the Advanced Land Air Research Model (ALARM). JTLS is distributed and supported on a worldwide basis by R&A through a CRADA. It is used for command post exercise support, analyses of contingency and operational plans, and examination of potential conflict situations. PAINT was used to develop counter terrorism plans, and ALARM was a research project designed to investigate organization structures, communications flows and movement of combat forces. Dr. Roland is Project Manager of the Naval Postgraduate School’s (NPS) Academic and Research support contract and the CINCPACFLT

Wargaming Division support contract. He is also the R&A corporate pilot.

Dr. Roland was a Professor in the Computer Science Department of the NPS where he designed and presented courses in computer science, command control, and information systems. He conducted research in artificial intelligence and advanced technology for management decision support systems. Dr. Roland acted as the project manager and director for acquisition, installation, and implementation of their first modeling, simulation and wargaming laboratory. His publications are available on request. Additional information is available concerning R&A at www.rolands.com. E-mail address: President@rolands.com.

BACK TO TOP

Information & Security. An International Journal
e-mail: infosec@mbox.digsys.bg

Applying Modeling and Simulation to Enhance National and Multi-National Cooperation

Ronald J. Roland

Keywords: Joint theater level simulation (JTLS), command and control, crisis management, national military command center (NMCC), computer aided exercises (CAX).

Abstract: The article presents case studies and practical applications enabling the readers to learn about military simulations and how they are used in defense and crisis management. It is the first in a series that will describe the application of modeling and simulation to enhance country-to-country, agency-to-agency and coalition-to-coalition cooperation and understanding and provides information on the availability of M&S applications, their potential benefits and associated cost. The content is relative to company news – updating the reader on the latest developments in the military simulation and training industry and assisting in developing a hypothetical plan for the implementation and use of various M&S applications. The coverage focus is generally on the command laboratory simulation but is defined in the context of crisis management center concepts.

Authors: **Judith Dahmann, Marnie Salisbury, Phil Barry, Chris Turrell, Paul Blemberg**
Title: **HLA and Beyond: Interoperability Challenges**
Year of issuance: **1999**
Issue: **Information & Security. Volume 3, 1999**
Hard copy: **ISSN 1311-1493**

HLA AND BEYOND: INTEROPERABILITY CHALLENGES

Judith DAHMANN, Marnie SALISBURY, Phil BARRY, Chris TURRELL and Paul BLEMBERG

Table Of Contents:

---

1. Introduction

The High Level Architecture (HLA) is a runtime software architecture for distributed simulation, which can be used by developers to create simulation applications. The HLA was developed in an environment that required the maximum amount of flexibility along with a minimum number of constraints. Simulations run the gamut from human-in-the-loop individual training simulators to highly aggregated, discrete event simulations that are used to exercise entire battle staffs. Beyond the training environment, the research and acquisition communities each have their own requirements for simulation applications. To address these diverse requirements, the HLA is designed to be flexible enough to support this diverse set of simulation systems in their unique environments and sophisticated enough to embrace the next generation of simulations. This broad-based environment sets the stage for the development of the HLA specification following the initial baseline definition of the HLA in March 1995.

In essence, HLA provides a framework and a set of capabilities to support the design and execution of simulation applications (federations) composed of multiple simulations and interfaces with live systems (federates). It is the task of the federation developer to apply these capabilities to meet the needs of their application. A key to this is ensuring that the selected federates *interoperate* in a way that technically and substantively meets the needs of the application. Consequently, for the purposes of this discussion, we consider interoperability from two perspectives: *technical interoperability* and issues of coherency and "fair fight" within the simulation domain or *substantive interoperability*.

As a technical architecture, the HLA addresses technical interoperability issues and provides users with support in understanding how to apply the HLA technical capabilities to create a working federation. Beyond the achievement of technical interoperability, there is a requirement to anticipate and address issues that impact on the ability of simulations to inter-operate in a logically meaningful manner. This paper identifies and discusses issues that affect interoperability in these two areas using the Federation Development and Execution Process (FEDEP) as a reference and guide.[1] The key interoperability issues are identified in a general way and are illustrated in the context of several battlespace applications. In the process, some activities that support HLA users to address interoperability challenges are outlined.

2. The Federation Development and Execution Process (FEDEP)

The Federation Development and Execution Process grew from the desires of the early HLA proto-federation development teams to organize and document the process for the creation and execution of an HLA federation. The FEDEP is a systems engineering process model that defines a generic framework for distributed simulation development. It is not unique to the HLA, but rather has been adapted to provide a common reference point for communication between federation development team members. The most common representation of the FEDEP consists of a graphical model, however, a supporting documented description is also available.

Recently the five step version of the FEDEP was expanded into six steps by segmenting the former Step 3, Design and Develop Federation, into two separate parts. This was undertaken in recognition of the complexities of these tasks, and the number of underlying events associated with each of them. In addition to providing a clear and concise introduction to federation development for HLA users, the FEDEP also offers a common foundation for defining functional overlays associated with the federation development process, such as, verification, validation and accreditation (VV&A), Security and Tools. The six steps in the process are:

Step 1: Define Federation Objectives

Step 2: Develop Federation Conceptual Model

Step 3: Design Federation

Step 4: Develop Federation

Step 5: Integrate and Test Federation

Step 6: Execute Federation and Prepare Results

Most recently, the FEDEP has been augmented with the addition of a checklist.[2] The idea behind the development of the checklists was to document many of the details that must be considered in the execution of the FEDEP and to group them logically within the activities to which they pertain. The checklists were developed based on the experiences gained through the development of several specific federation applications. In general the items on the checklists are applicable to all federation efforts, but of course, they are tailorable as is the FEDEP itself.

The concept behind the FEDEP then is to provide a structured methodology to address the issues associated with the development and execution of an HLA federation. As such, it provides a useful framework for the discussion of simulation interoperability issues.

3. Technical Interoperability

Often when one discusses interoperability, the focus is on technical interoperability. Technical interoperability is achieved when the federation is integrated, and individual federates are passing data in accordance with the federation object model (FOM). This involves the use of common standards, compatible interfaces and coordinated data structures.

Technical interoperability is the capability of federates to physically connect and exchange data in accordance with the FOM. Elements include the following:

- Standards compatibility
- Hardware compatibility
- Time management coordination
- Coordinated use RTI Services
- Security issues (if applicable)

Appropriate use of *standards* is a key element of technical interoperability. Complying with the HLA standard, as an individual federate or as a federation,[3] provides assurance that the specifications of the architecture have been correctly implemented, and hence basic capabilities for operation are met. It is a minimum essential capability for developing a federation. Other standards that contribute to technical interoperability are those for operating systems, languages, and network protocols.

Hardware compatibility is another basic essential technical component to federation development and execution. Physical interoperability, or the actual connection of one system to another, is one level of hardware interoperability while systems capabilities and capacities are another. For example, differences in the capabilities of network cards may cause performance problems within a federation that fall under the heading of interoperability. Consider the case of a 10BaseT network card in a

100BaseT LAN environment. These issues generally come to light during Step 5 of the FEDEP. The Federation Execution Planner's Workbook [4] is one of the mechanisms available to assist federation managers in identifying and resolving these issues before they become problems. Among other information, the workbook provides a place to capture information about the computing environment in which the federation is executed, and therefore provides insight into potential incompatibilities. Another hardware interoperability problem is the issue of data formats and byte-ordering, and the issue of big endian versus little endian. These issues are well documented and will not be discussed here, but need to be addressed in the federation development process during Step 4 of the FEDEP, Develop Federation.

The use of *time management* is a key element of technical interoperability in a federation. The mechanics of interoperability between time regulated and unregulated simulations must be considered and resolved. The HLA was designed from the outset with the capability to facilitate interactions among dissimilar simulations. The Time Management Services provided by the Runtime Infrastructure provide the means of solving the problem, but the correct use of these services is left to the federation development team. Failure of the federates to use the services correctly can result in a federation that stalls or never runs forward in time.

There are several *RTI Services* that are intended to support and apply to the entire federation, not just individual federates. Examples of these are the synchronization point services, save and restore services, and data distribution services. Uncoordinated or inaccurate use of these services within a federation can cause catastrophic failure of the federation execution. Use of these services must be addressed by the federation development team during FEDEP steps 3 and 4, Design Federation and Develop Federation. For example, will synchronization points be employed; if so where and how? If synchronization points are used, then some or all federates must be able to respond to them. If not, is a surrogate, such as the Federation Management Tool, available to respond on their behalf?

Security is the last issue to be addressed under the heading of technical interoperability. Data encoding and decoding, levels of security and methods for bridging secure and unsecured networks must all be addressed. Security considerations can be identified as early as Step 1 of the FEDEP and are addressed throughout the execution of the process. A security overlay to the FEDEP has been developed and is available to assist federation management teams with these issues.[5]

4. Substantive Interoperability

Resolving technical interoperability issues insures that the federation will run, but says nothing about the adequacy of the federation to accomplish its mission. Unlike technical interoperability where something works, or it does not, substantive interoperability is driven by the needs of the federation and has to be addressed by each federation in a federation specific way. Thus these issues are much less obvious, and therefore more difficult to solve in a general way. The level of interoperability required depends in part on the desired outcome of the federation, the capabilities of the individual federates involved, and the degree to which consistency must be maintained to achieve the desired results. These considerations are different for almost every implementation, and therefore, the standardized methodology offered by the FEDEP becomes even more important as a means of focusing on the issues. The FEDEP provides a structured methodology for the development of the federation and along with the Checklists identifies the appropriate places in the process where these issues should be considered. The remainder of the paper will be devoted to the identification and discussion of coherency and "fair fight" issues that transcend technical interoperability.

4.1. Interoperability among Entities

When you form a federation, you are technically linking the federates, but you are logically linking the entities represented inside of each of the federates. To have a meaningful federation, the set of entities represented across the federation work together in a manner consistent with the needs of the federation application. In effect, in an HLA federation, you are creating an end-to-end model by reusing selected representations from the participating federates. The data in the FOM defines the exchanges among the federates, but the entity representations and their combined behaviors define the new simulation application supported by the federation.

The purpose of the federation is the key definer of this process. Defining the representational needs of a federation is the starting point for creating one that meets the needs of the user/sponsor. A good conceptual analysis during Step 2 of the FEDEP can not be overemphasized. During the Conceptual Modeling phase of the FEDEP the federation development team and sponsor develop the story line describing the sequence of activities, that, when executed in the federation, will achieve the federation objectives. This usually involves the creation of a scenario and a conceptual analysis. The scenario defines the activities and relationships that will exist among the entities, while the conceptual analysis captures a "real world" authoritative description of the entities and actions that must be represented in the federation. This analysis will influence the ultimate selection of federates as well as the development of the Federation Object Model (FOM). Selecting federates with the right representations for the purpose of the federation is key to substantive interoperability

While the specific areas which need to be addressed in any application will be defined by the needs of the federation, the type of issues to be considered include:

- *Entity level of representation*: Does the level of representation of entities included in the federation support the federation conceptual model, and as a group logically ‘fit’ together?

All entities do not necessarily need to be represented at the same level, but interactions between entities (e.g. individual weapon systems fired against composite targets) need to be appropriate for the needs of the federation.

- *Entity attribution:* Do the entities in the federation incorporate representation of the key attributes salient to the federation purposes in a sufficiently consistent manner to meet the federation needs?

Issues of completeness and consistency of representation across the federation are important primarily in the selection of federates. In Step 3 of the FEDEP, the federation manager selects federates and assigns responsibility to represent attributes of classes of objects to them. An important part of this process is the assessment of the ability of the participating federates to represent the critical characteristics of entities needed to meet the federation mission.

- *Entity behaviors*: Do the behaviors of the individual entities reflect the needs of the federation conceptual model and do they 'work' together logically to meet the needs of the federation? This includes the algorithms used in the simulations to compute the effects of one entity on another.

For example, for simulations to be considered interoperable in supporting an application involving battlespace outcomes, the effects of weapons must be consistent and realistic from both sides of a weapons engagement. Consider two simulations, A and B. Let simulation A represent ground forces in a federation and simulation B represents the air forces. Suppose that aircraft modeled in simulation B attack a target in simulation A with air to ground missiles. Suppose further that simulation A returns fire with surface to air missiles. Which simulation computes the attrition in each case? Is the calculation mathematically correct and from whose perspective? A key to interoperability is to determine if the weapons effects of each of the simulations on the objects owned by the other are realistic.

It is important to note that this assessment must include both the behaviors of entities that will be exposed to the federation (HLA objects) but also the behaviors of relevant entities that are wholly modeled inside an individual federate. For instance, consider the case of a federation designed to assess comparative capabilities of different sensors in a particular scenario that includes weather effects. In this case, it is important to ensure that the effects of weather are represented in the different federates, and these effects are consistently modeled across the set of sensors whose capabilities are under assessment.

- *Temporal resolution*: Each simulation computes changes in state of the entities it represents at some resolution of simulation time. While all entities in a federation application do not have to resolve time at the same rate, it is important that the overall end-to-end model has incremental state changes computed in a fashion commensurate with the needs of the application.

As an example, consider two simulations. The first is focused on ground combat, the second on air warfare. Suppose the aircraft in the aviation simulation attacks a ground target that defends itself with surface-to-air missiles (SAMs). Not only must the three dimensional locations of the aircraft, the SAM and the SAM site be consistent, but the time steps for the individual simulations also need to be representative of the activities being simulated. In the example, consider the situation where the attacking aircraft is flying at 480 knots which translates to eight miles per minute. If the simulation modeling the SAM systems searches for air targets at one-minute time steps, the attacking aircraft could directly overfly the SAM and possibly destroy it in between timesteps. In other words, at the first time step, the aircraft would be out of detection range of the SAM. At the second time step, the aircraft will have attacked the SAM and destroyed it without the SAM ever having the opportunity to acquire the aircraft. This would not accurately model the real world at a behavioral level.

- *Spatial resolution*: As with time, entity behaviors are represented with spatial 'granularity'. Here too, the spatial resolution of entity behavior across a federation does not necessarily need to be common but needs to logically fit together in a way that supports the federation application.

The importance of spatial resolution can be illustrated with examples analogous to that cited for temporal resolution above. Consider the example of a federation designed to examine comparative effects of targeting technologies or technologies that improve the abilities of sensors to identify targets. In this federation, the spatial resolution of entities will be key to the results of the federation. If the representation of the terrain is too coarse, then entities may occupy locations at fixed intervals from each other and may appear in regular patterns. This may make target detection and identification unrealistically easy for the targeting technology. If the terrain representation used for the entities is more detailed than that used by the simulation of the targeting technology, then the movement of entities may be indiscernible and may prevent valid target detection or identification.

Considerations of this type are key in the selection of federates to meet the needs of a federation. "Choosing your partners carefully" is the most important part of designing a meaningful federation. Good federate selection depends on a solid federation conceptual model (FEDEP Step 2) that identifies the appropriate levels of resolution of key entities, the salient attributes and behaviors, the particular aspects of entities and behaviors relevant, and the granularity of temporal and spatial resolution appropriate to accomplishing the federation purpose. Given a good statement of needs of the federation, the federation designer can evaluate the capabilities of candidate federates.

The success of the federation designer is also dependent on the ability to understand the capability of individual federates. Unless information on candidate federates is available, it will be difficult to assess their appropriateness to meet specific federation needs. The SOM for a federate is of limited use in this exercise, since by design it addresses only the data available for exchange in a federation. Here the interest is in the internal representations in a federate whether or not they are exposed at runtime. Given the lack of documentation on simulations and the ad hoc nature of this documentation, today this task typically involves person to person interaction between a federation manager and a person knowledgeable about the federate.

Finally, the federation manager needs to consider the composite suite of federation members. Each federate may meet the needs of a piece of the federation problem. However, the real question is whether the federates as a group work together to meet the needs of the federation as a whole. That is, whether the end-to-end model represented in the federation is appropriate to the federation needs.

4.2. Contextual Effects on Interoperability

In any simulation, the represented entities are situated in a context that affects their behavior as much as the interaction with other entities. When bringing together entities represented in different federates it is very important to ensure that the contexts of these entities are consistent in the areas of importance to the federation. In battlespace simulations, the most critical contextual effects are often environmental (natural and manmade). Depending on the nature of the simulation, there can be other effects whose impact is felt across the federation such as communications and the electromagnetic atmosphere.

In addition to the contextual factors themselves, the *effects* of these factors on entities in the battlespace scenarios must be sufficiently consistent to meet the needs of the federation. Common environmental representation is an important starting point, but beyond this, the effects of terrain, for instance, on line-of-sight calculations of simulated platforms represented in different federates can be crucial in creating a coherent federation.

In this section we will focus on the environment because of its overriding importance in most battlespace applications.

As discussed in [6], for multiple simulations to interoperate in the same domain they must have a sufficiently common, correlated view of the *environment* to support the needs of the application. The development of authoritative environmental representations is time consuming and costly, describing all of the relevant elements and environmental events and interactions that are expected to occur in an area of interest. Additionally, environmental representation involves description of the geometric and topological relationships between data objects. To create and sustain realism and accuracy, all federates should have a consistent perception of the environment.

Environmental representation consists of both the visual and non-visual aspects of the environment, and is composed of digital representations of the atmosphere, space, ocean, and terrain. If germane to the simulations, other aspects of the electromagnetic spectrum may also be represented such as infrared radiation and radar reflectivity. Additionally, objects on the battlefield such as trees, buildings, bridges, and highways, etc. may also need to be represented.

Previously, consistency in environmental representation was achieved by developing a mapping between the individual environmental representations of simulations that were required to interoperate. This mapping rectified disparate coordinate systems as well as resolving differences in fidelity and interpretation between the different environmental representations, which is critical to ensure accuracy of weapons and intelligence effects. In general, this type of point to point mapping was based upon proprietary database formats that required the development of customized software to convert the source data into a format that is useable by the simulation. The greater the number of different database formats involved the more, the greater the cost to achieve consistency, if even possible.

The task of ensuring a consistent environmental representation can be decomposed into several logical pieces. First, the environmental data must be drawn from a common authoritative source. Identifying a common authoritative source and generating a consistent environmental scenario eliminates a significant potential variability at the beginning of the process.

The next step is to establish access to the integrated data sets. There are numerous source databases. One approach to provide access to these data sets is to write software that will convert from native format(s) into the format of the simulation database generation system. Depending on the type and variability of the sources involved, this could be a costly and time-consuming process. The Synthetic Environment Data Representation and Interchange Specification (SEDRIS) project has developed a common interchange mechanism with supporting transmittal formats to provide access to all environmental representations through a standard interface. The transmittal format fully supports a

Data Representation Model (DRM). A Data Coding Standard (DCS) provides enumeration for the elements within the DRM.

The DRM and the DCS are logical constructs. The SEDRIS Transmittal Format (STF) is the physical construct that defines the order and physical formatting of the data. To use the STF one creates a SEDRIS transmittal by accessing the *Write Application Programmer Interface (API)* definition within the SEDRIS Interface Specification. The *Write API* takes data content from the native format of the data source as expressed in the DRM and creates a transmittal.

Extraction of environmental data from a transmittal is the function of the SEDRIS *Read API*. As defined in the Interface Specification, the *Read API* consists of a number of functions that pull data from a SEDRIS transmittal. To get the data into the native format of a particular simulation, converter software must be written to interface with the *Read API*. The converter software is simulation specific and must be maintained as native data representations evolve. Designing and implementing the converter software is one means for addressing the fidelity issues that must be resolved between the source data and simulation runtime requirements. Converter software can also use utilities in the SEDRIS *Read API* support libraries to perform requisite coordinate conversions and/or datum transformation.

The advantage of the SEDRIS approach is that a common interface to source data is provided along with the tools to use this interface. Each federate now only needs to write converter software to one standard. Content consistency is achieved through the use of a complete and unambiguous interchange mechanism. The cost of getting data into the simulation is reduced since only one set of conversion software is required to access a complete range of environmental and 3-D model data. Use of SEDRIS and its supporting capabilities aid the federation developer in creating common, or at least consistent, environmental representations across a federation as a context for entity representations.

5. An Example: Command, Control and Communications (C3) Interoperability

In this section we review several notional use cases of interoperability issues arising in applications typical of defense users, those which incorporate consideration of command and control. The representation of command, control, and communications within a federation can pose several interoperability challenges. In this section, we describe several challenges and solutions that have been applied in other federations. Although the problem space has been divided into categories in this section, the categories are quite inter-related.

5.1. Command Organizations and C2 Process

The first element to examine is the portrayal of command organizations and the command and control (C2) process within and outside the federation. Depending on the fidelity of the simulation and the intended uses for it, the representation of command organizations and the C2 process can vary widely. The features of the individual simulations must be well understood before the issues related to interoperability within the federation are examined.

The features of a simulation related to command organizations and the C2 process may be characterized by the answers to a few questions:

- Are the physical elements of command units (e.g., command posts) represented?
- Are the pre-planned actions of combat units directed or initiated by logic within the simulation or by intelligent human operators?
- Does the simulation model the cognitive decision-making functions of commanders or command units and to what extent?
- What impact does the collection and processing of situation awareness data have on the cognitive decision-making functions simulated? Are cognitive decisions made using perceived data about the battlefield situation, or ground truth data stored and calculated within the simulation?
- What battlefield activities, such as attrition or communications, affect the pre-planned or reactive actions of combat units?
- What battlefield activities, such as attrition or communications, affect the quality, completeness or timeliness of output of the cognitive decision-making functions?

Based on the answers to these questions for each simulation in the proposed federation, one can start to characterize the situation for the federation. The following list includes a set of examples:

- One or more federates has very detailed models of C2 command posts and C2 processes, while other federates have more simplistic models or none at all.
- One or more federates contain brittle models of cognitive decision-making, where the decision-making capability breaks down easily when confronted with a previously unknown situation, causing combat units to continue blindly and illogically forward or to stall and do nothing.
- One or more federates executes cognitive decision-making processes using ground truth data while other federates use perceived knowledge of the situation.
- One or more federates contain invulnerable models of command posts, where the C2 organizations, equipment, or processes cannot be influenced by battlefield outcomes.

The interoperability problems that arise from these situations can be described as "fair fight" issues. Here are some examples:

Potential Fair Fight Problems

- The actions and decisions of a unit or force whose cognitive decision-making is based on perceived data about the battlefield situation may be less effective or accurate than those of units whose cognitive decision-making is based on perfect knowledge of the battlefield situation.

- The results of specific offensive actions or campaigns against the command organizations and C2 processes of an opposing force will not be effective if the opposing force is modeled with no vulnerabilities.
- The actions and effectiveness of a unit or force with brittle model(s) of cognitive decision-making may be very adversely affected if it gets into a paralyzed state where it can't make a coherent decision and execute protective actions.

5.2. Command and Control Information Exchanges

Information and data are the resources that drive the C2 process. The Federation Object Model and the RTI services make the technical part of information exchange in a federation straightforward, but other challenges remain. The three key cognitive components of a coherent and successful information exchange are as follows:

1. Something meaningful to say
2. Someone to say it to
3. A recipient that can interpret what is said and react to it

Disparities between the representations of command organizations and the C2 process in different federates will definitely create interoperability problems when those simulated command organizations attempt to exchange information. These problems may be "fair fight" issues or technical incompatibilities that may cause failure during federation execution. Some examples are provided below:

Potential Fair Fight Problems:

- A federate with a detailed model of the C2 process may require information from other command organizations that cannot be provided. The absence of this information may result in less effective decisions and poor outcomes for the simulated units in the first federate.
- A federate with a detailed model of the C2 process may send sophisticated orders to command organizations in other federates assuming that the information will be received and acted upon correctly. If the receiving federates lack the capability to receive and interpret the orders, the actions carried out by those units may be ineffective and detrimental to the simulated units or command organizations in the first federate.

Potential Technical Incompatibilities:

- A federate with a simplified model of the C2 process or no model of the C2 process will lack item #2 "someone to say it to" and item #3 "a recipient that can interpret and react". Information that is sent from another federate to this federate will result in inappropriate action or no action.

- A federate with a simplified model of the C2 process may lack the ability to generate meaningful information (item #1) and pass that information to other federates.

5.3. Communications Effects

In real life many types of communications techniques, devices, and networks are used to move information around the battlefield. Building and maintaining a reliable communications infrastructure of sufficient bandwidth is a critical part of military operations. The degree to which the means of communication are modeled varies widely from one simulation to the next.

The features of a simulation related to its model of communications may be characterized by the answers to a few questions:

- Is the physical communications equipment represented? Can it be damaged and rendered ineffective by ordnance or electromagnetic means?
- Is the probability of successful information delivery over the communications infrastructure affected by the terrain, the weather, or the electromagnetic environment?
- Is the communications model flow-based or item-based? That is, does the communications model calculate the number of messages that flow from point-to-point in a given time period or does it provide a probability of delivery for each specific information exchange?
- Does the communications model account for resource contention on the communications infrastructure?

A set of federates with different approaches to communications modeling can lead to "fair fight" problems such as the following:

Potential Fair Fight Problems:

- A federate with a detailed model of communications equipment and networks may degrade the completeness and timeliness of information exchanged between simulated units or command organizations within the federate or information arriving from another federate. This degradation may lead to delayed decision-making or lower quality decisions within the federate.
- A federate with a detailed model of communications may also have a more detailed model of the impact of the terrain, the weather, and the combat actions of the opposing force on the communications infrastructure. In other words, the federate may show degraded combat effectiveness due to failures of the communications infrastructure. A federate that has a low fidelity model (or no

model) of the communications infrastructure will not be affected by opposing force actions directed against it.

5.4. Possible Solutions

The solutions applied in existing examples fall into four basic categories:

1. Augment the federation of automated simulations with intelligent human operators.
2. Add functionality to the existing federates
3. Add one or more federates to bridge the coherency gaps between the existing federates or to provide a consistent modeling service to the entire federation.
4. Rescope federation expectations or factor out inconsistencies across the federation in interpretation of federation results

Augment with Intelligent Human Operators

One common way to bridge the gaps among inconsistent models of command organizations and the C2 process or to solve information exchange incompatibilities is to insert an intelligent human in the loop. The human receives and interprets the information generated by one federate and then creates a set of information that can be conveyed to one or more other federates. The intelligent human operator may perform any or all of these functions:

- *Translation:* Translate information from the set of terms generated by one federate to another set of terms that can be interpreted correctly by another federate.
- *Orders decomposition:* Interpret the orders generated by a command organization (simulated or live) at one level of the command hierarchy into a set of orders that can be executed by a lower level unit.
- *Aggregation:* Compute or abstract a higher-level view of a given set of data.
- *Degradation:* Intentionally changing the data generated by one federate before passing it on to other federates to introduce an appropriate level of error or insufficiency.
- *Augmentation:* Creating appropriate data that is not otherwise available in the federation, but is required by other federates.

Add Functionality to Federates

An obvious, but significant option is to extend the capabilities of a federate to make it match more closely the capabilities of the other federates. It may be quite difficult to add the necessary functionality to an existing federate. This approach could lead to a major redesign or rewrite of the software of the original federate.

Add Federates

New federates can be added to the federation to provide the capabilities that are missing or to provide a consistent model of a specific battlefield activity or effect. This approach has been used in both ways. One example is the use of artificial intelligence technologies to create models of specific command organizations that can interpret and react to orders and situational awareness data and execute the C2 process to produce "orders" or directives that can be interpreted by simulated units in another federate. The second way is the use of a federation-wide server to provide a specific algorithm or service to the federation to achieve consistent results. For example, a communications effects server has been used to consistently apply terrain or weather effects to information exchanges over communications devices within a federation.

Rescope Federation Purposes or Factor out Federation Inconsistencies

When adjustments such as those described here are not viable, there are two other options that the federation manager might consider. These are typically considered when resources (i.e., funding or time) don't allow for changes in the federation, but the users still see value in the federation execution.

- Ignore the incompatibilities to the extent that they don't cause the federation execution to fail, but factor out inconsistencies in interpretation of the federation results. Recognizing that simulations never provide a full representation of all desired capabilities, this approach may be workable in some circumstances.
- Eliminate modeling features of particular federates to get the federation down to a level (but lower fidelity) playing field. This may provide the consistencies needed, but limit other things that might have been of interest to the federation user. This approach requires a rescoping of the federation purposes to achieve some portion of the federation goals. Only, the federation manager can make tradeoffs such as these in conjunction with the federation user.

6. Summary

In summary, this paper has reviewed a range of technical and substantive interoperability issues to be addressed by a federation developer in the use of the HLA.

Technical interoperability challenges that must be addressed in order to configure and run a federation; these are basic to the use of HLA and are a necessary prerequisite to application of HLA to user problems. These include conformance with standards, hardware interoperability, coordination of time management coordinated use of RTI Services, and security.

While important, technical interoperability does not guarantee the creation of a meaningful federation application. Federation designers must address these issues of substantive interoperability in the design of the federation since the nature and degree of interoperability needed is driven by the purpose of the federation application.

The federation conceptual model, which lays out the representational requirements of the federation, is critical in this process. It provides the criteria for selection of federates based on the characteristics of their representations and their ability to meet the needs of the federation application. Assessing characteristics of the entities in each federate including their level of representation, their attributes and behavior, and their temporal and spatial representation, is the heart of this process. This assessment is done both in terms of the extent to which the entities individually address federation needs as well as the extent to which the set of entities fit together to meet the end-to-end needs of the federation application.

Beyond the entity characteristics, the context in which these entities operate, including the environment and other federation-wide effects, need to be assessed from the perspective of the federation needs. It is important to ensure that the environment and other contextual factors are consistently represented across the federation and the interactions between the entities and these contextual factors are sufficiently consistent to meet the needs of the application.

Successfully addressing these substantive interoperability issues is the responsibility of the federation manager and federation designer. Support in this process is beginning to appear in several forms. The FEDEP and its associated checklist provide a starting point for this process. The Federation Execution Planner's Workbook provides aid in addressing technical interoperability issues. SEDRIS and its supporting capabilities can assist in addressing important environmental interoperability issues. In several cases (weather, for instance) the idea of federation servers is growing in an effort to provide consistent representation of dynamic federation-wide effects at runtime.

The efforts underway to develop processes for verification, validation and accreditation (VV&A) of federations are addressing many of the issues raised in this paper. In essence, building a federation that incorporates representations appropriate to the needs of the federation application is the heart of the VV&A problem. Further, work in the area of fidelity [7] is attempting to more precisely define some of the issues raised with respect to entity characteristics; both those needed by an application and those provided by a simulation.

Finally, as HLA become more widespread there greater attention is paid to how it is being used, rather than how to use it. This shift in emphasis highlights the importance of raising awareness of interoperability issues and developing techniques and solutions for addressing them.

References:

1. HLA Federation Development and Execution Process (FEDEP) Model, Version 1.4 (June 1999).
2. HLA Federation Development and Execution Process (FEDEP) Checklists,

Version 1.4 (Draft, May 1999).

3. HLA Compliance Checklist, Federate, Version 1.3 (May 1998).
4. Federation Execution Planner's Workbook, Version 1.3r3 (May 1999).
5. Security Overlay for the FEDEP, available at http://hla.dmso.mil/hla/federation/security/.
6. U.S. Army Simulation, Training and Instrumentation Command. "SEDRIS and the Synthetic Environment Domain – Volume 1 of the SEDRIS Documentation Set" (March 1998).
7. Gross, David et al. "Report from the Fidelity Implementation Study Group." Paper 99S-SIW-167, presented at the Spring Simulation Interoperability Workshop (March 1999).

JUDITH S. DAHMANN, Ph.D. is the Chief Scientist for the U.S. Defense Modeling and Simulation Office. She has led the development of the High Level Architecture. Address: DMSO, Suite 500, 1901 N. Beauregard St., Alexandria, VA 22311-1772, tel.: (703) 998-0660. E-mail: jdahmann@msis.dmso.mil.

MARNIE SALISBURY is a Principal Engineer at The MITRE Corporation. She has seven years of experience in battlefield simulation and interoperability, including 15 months as Technical Director for the DARPA STOW program. She currently leads the HLA cadre outreach team. Address: The MITRE Corporation, 1820 Dolley Madison Blvd., McLean, VA 22102, tel.: (703) 883-7064. E-mail: marnie@mitre.org.

PHILIP S. BARRY is the Deputy Division Chief for the Technology Applications Division at DMSO and a Lead Systems Engineer for the MITRE Corporation. Dr. Barry has a Ph.D. in Information Technology and a M.S. in Systems Engineering from George Mason University, and a B.S. in Aerospace Engineering from the University of Virginia. Address: The MITRE Corporation, 1820 Dolley Madison Blvd., McLean, VA 22102, tel.: (703) 883-7826. E-mail: pbarry@msis.dmso.mil.

CHRIS TURRELL is a Senior Simulation Analyst with AB Technologies, Inc. of Alexandria, VA where he is the program lead for the Defense Modeling and Simulation Office HLA Tools and Services development efforts. He is actively involved in the DoD Joint Technical Architecture development process and participates in related IEEE and SISO standards development activities. Address: AB Technologies, Inc., 1901 N. Beauregard St., Suite 400, Alexandria, VA 22311-1772, tel.: (703) 824-3457. E-mail: cturrell@msis.dmso.mil.

PAUL BLEMBERG is a senior analyst at AB Technologies. He is the liaison between the Modeling and Simulation Information Analysis Center (MSIAC) and the Technology Application Division of the Defense Modeling and Simulation Office (DMSO). He has a BS in Psychology from Northern Michigan University. Address: AB Technologies, Inc., 1901 N. Beauregard St., Suite 400, Alexandria, VA 22311-1772. E-mail: pblember@msosa.dmso.mil.

BACK TO TOP

Information & Security. An International Journal
e-mail: infosec@mbox.digsys.bg

HLA and Beyond: Interoperability Challenges

Judith Dahmann, Marnie Salisbury, Phil Barry Chris Turrell and Paul Blemberg

Keywords: HLA, Federation Development and Execution Process (FEDEP), Interoperability

Abstract: The High Level Architecture (HLA) provides a foundation for technical interoperability among disparate simulations, simulators, live systems, and simulation support tools. Achieving technical interoperability is a significant first step, but it is only part of the interoperability equation. This paper describes other interoperability challenges that federation designers and managers face. These challenges relate to the differences between modeling techniques employed by different simulation federates and the logical interpretation of data shared among federates. Using the Federation Development and Execution Process (FEDEP) as a guide, we identify the key areas where federation developers address simulation interoperability as they apply HLA to their domains particularly in battlespace simulation applications and cite potential sources of support for addressing these issues.

Author: **Paul J. Graziani**
Title: **Commercial-Off-The-Shelf Software Becomes Mission-Critical to Success and Cost-Effective Space Missions**
Year of issuance: **1999**
Issue: **Information & Security. Volume 3, 1999**
Hard copy: **ISSN 1311-1493**

COMMERCIAL-OFF-THE-SHELF SOFTWARE BECOMES MISSION-CRITICAL TO SUCCESS AND COST-EFFECTIVE SPACE MISSIONS

Paul L. GRAZIANI

Table Of Contents:

Governmental space budgets are tightening. The era of huge space programs with virtually unlimited budgets is over. Meanwhile, the markets for commercial services delivered by satellite continue to grow at 17 % CAGR,[1] driven by increased access to launch services, lower-cost satellites, and global-scale business opportunities. Competition, though, means that cost-effectiveness is more critical to the success of a mission than ever.

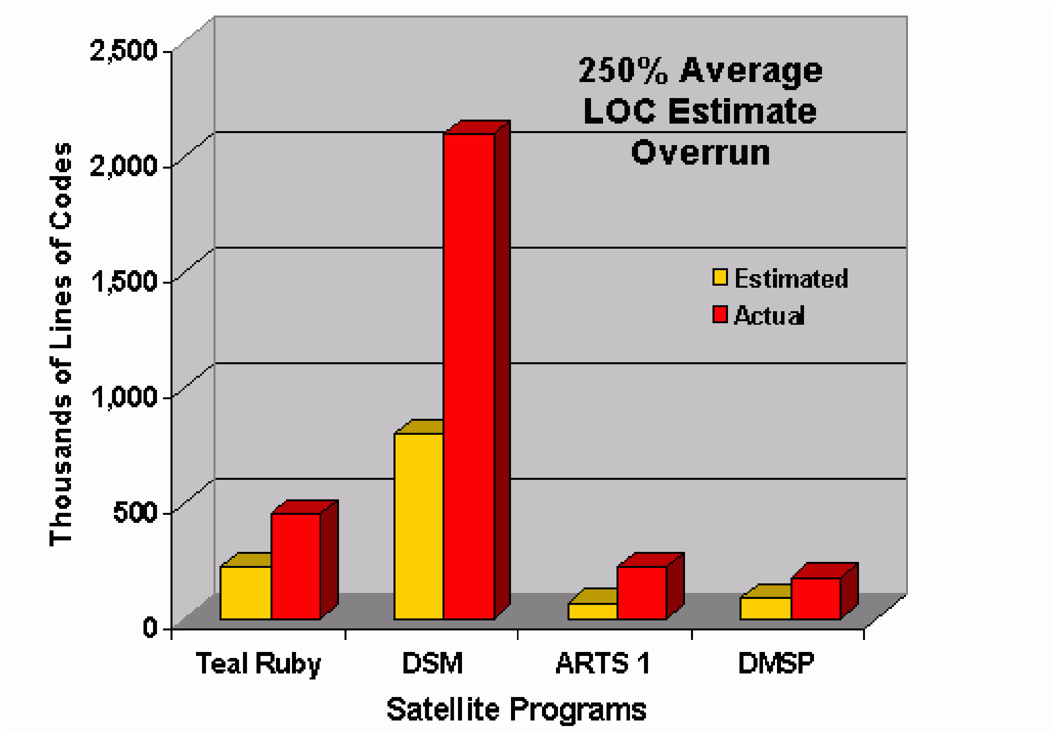
Many organizations can no longer afford to develop mission-specific software in-house. Commercial satellite software that can be used for multiple missions is playing a growing, cost-saving role in enabling not only successful but economical space activities. Commercial off-the-shelf (COTS) software that is stable, accurate, and well-supported is key in all phases of a satellite's life—from pre-launch analyses to ground support through to graveyarding.

Software Development Costs

Historically, satellite programs have been funded to develop custom software solutions from scratch. It has been identified that software development is the largest cost-risk factor during the development of a new satellite system. This is driven by two factors:

- Lines of code underestimated
- Technological challenges – programmer productivity decreases

The chart below identifies four large satellite programs which all significantly underestimated the scope of their software requirements which resulted in significant cost overruns and failure to meet delivery schedule.



Typical satellite programs have shown a Line-Of-Code (LOC) overrun of approximately 2.5X – to compound the software development risk, programmer productivity typically falls to about 28 % of estimated rate (3.5X slower than estimated). Due to competitive forces, aerospace contractors are forced to bid/estimate programmer productivity at a rate of 300 LOC per month; however, the average productivity turns out to be 85 LOC because of unforeseen technological challenges as well as unrealistic expectations.

Software Cost-Growth Factor

- 250% growth in LOC estimate
- 350% slip in productivity
- 250% X 350%

880% cost growth in satellite software projects

Another unpredictable cost associated with in-house or built-from-scratch software solutions is that follow-on maintenance of the software project is often not considered within the initial funding of the program, key developers may leave the project, or unneeded contractors may become entrenched into the program office.

COTS software has proven that it can meet the unique needs of each space program but not require the years to create and millions of dollars to produce. COTS software has also proven that it has the technical depth and diversity of capabilities to handle risky situations where new ground is broken, both figuratively and literally. This increasing reliance on COTS solutions demands better and more robust products that can address a wide variety of analysis and planning needs. Customers are looking for an economical solution that remains technically accurate and easy to use.

The IMACCS Project

NASA's Mission Operations and Data Systems Directorate (MO&DSD) created the Reusable Network Architecture Interoperable Space Science, Analysis, Navigation, and Control Environment (RENAISSANCE) project to re-engineer the process used to build and operate spacecraft ground support systems. The primary goal of the RENAISSANCE project was to achieve cost-effective and flexible development and operations based on workstation/LAN/file server technology and reusable hardware and software building blocks.

The benefit of using COTS software in the RENAISSANCE effort was the construction of reusable, generic, software "building blocks" based on legacy software. In theory, new missions could then select among these building blocks to build systems in an object-oriented manner.

CSC, a contractor to NASA, suggested that an approach emphasizing the use of COTS products rather than software developed in-house would be far more cost effective. Although it was commonly believed that the COTS market had yet to reach sufficient maturity for most ground support functions, some key NASA personnel backed CSC's proposal. As a result, the Integrated Monitoring, Analysis and Control COTS System (IMACCS) project was approved.

Based in part on prior positive experiences with COTS satellite analysis software systems, CSC was confident that COTS systems could be utilized to produce a robust system that could meet or exceed the current ground support system in terms of functionality, implementation time, and implementation costs. The IMACCS challenge was to assemble a COTS-based system that could mirror current ground support, from data decommutation and conversion through spacecraft monitoring and commanding. IMACCS would also be responsible for off-line activities such as orbit and attitude determination and predictive products.

To demonstrate the feasibility of a COTS system, CSC was tasked to create a proof-of-concept system that performed all current ground support tasks. With a 90-day turnaround time constraint, CSC chose the Solar Anomalous Magnetospheric Particle Explorer (SAMPEX) satellite as the test vehicle for the proof of concept.

SAMPEX studies the energy, composition, and charge states of particles. It also closely monitors the magnetospheric particle populations that plunge occasionally into the middle atmosphere of the Earth, thereby ionizing neutral gases and altering the atmospheric chemistry. A key part of SAMPEX is to use the magnetic field of the earth as an essential component of the measurement strategy. The Earth's magnetic field is used as a giant spectrometer to separate different energies and charge states of particles as SAMPEX executes its near polar orbit. SAMPEX is thus a very versatile mission capable of scientific investigations in astrophysics, solar physics, magnetospheric physics and atmospheric physics. Figure 1 is a picture of the SAMPEX satellite orbiting Earth.

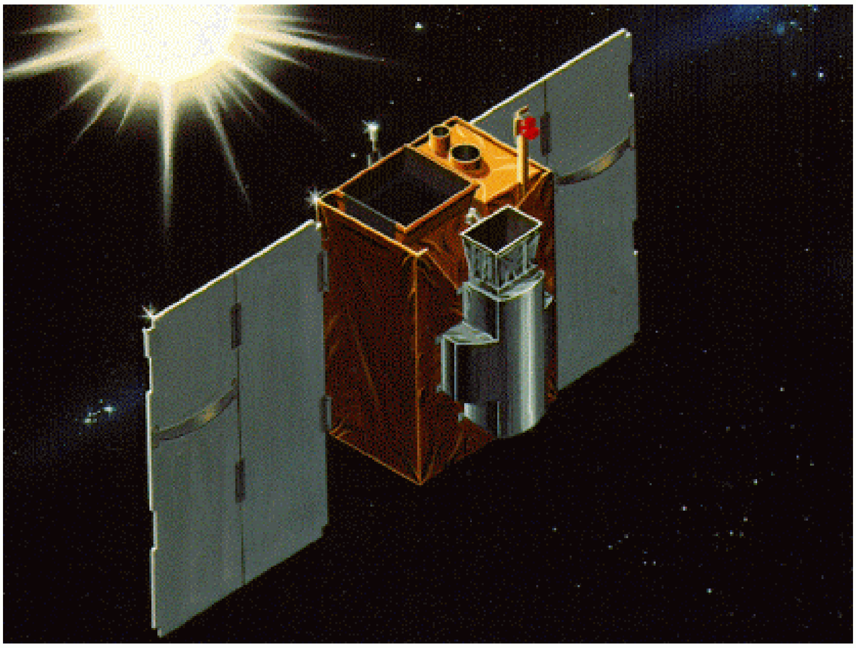


Figure 1: SAMPEX Satellite

Because SAMPEX is a relatively small program and is currently operational, it was ideal for the project. In addition, SAMPEX produces telemetry data in CCSDC format, a relatively new format that is fast becoming popular. The IMACCS project consists of several COTS systems. Figure 2 summarizes the IMACCS structure

IMACCS used the COTS software to perform orbit determination and propagate ephemeris. One module was used to accelerate access determination and group ground stations together for access reporting purposes. Other tasks performed by the IMACCS system included predicting potential ground station contact times, lighting times, times of passage through the South Atlantic Anomaly and other electron contamination regions, and node crossings. With this software, these prediction exercises could be easily viewed in the 2D map window as ground and/or orbital tracks. The software was also used to predict the position and velocity data delivered to the ground stations and control centers for uplink to the spacecraft.

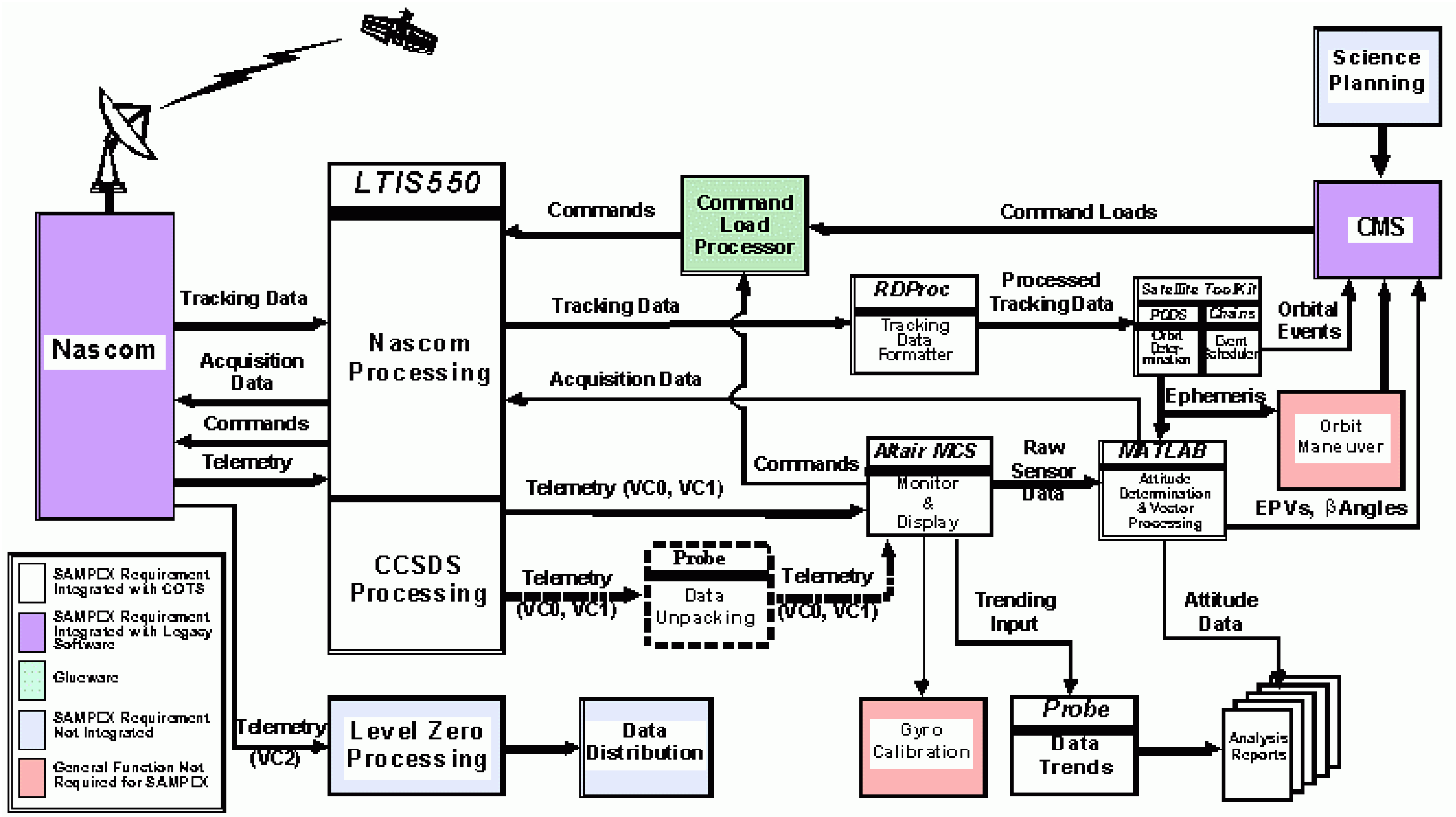


Figure 2: IMACCS Structure

Defining Regions of Interest

The South Atlantic Anomaly is a variation in the Earth's magnetic field that allows cosmic rays and charged particles to reach deeper into the atmosphere. This can interfere with communications among spacecraft and aircraft, and may also corrupt scientific data gathered by the instruments onboard. CSC created an Area Target to represent the approximate size and shape of the Anomaly's widest boundary so that passage times through the anomaly could be computed and delivered to the Flight Operations Team (FOT). Figure 3 shows the area target as represented in the software's map window.

Area targets were also defined to represent regions that may or may not be of use to the SAMPEX satellite. Based on this data, control center personnel could determine whether instrumentation could be turned off to conserve power when the satellite traveled over these regions.

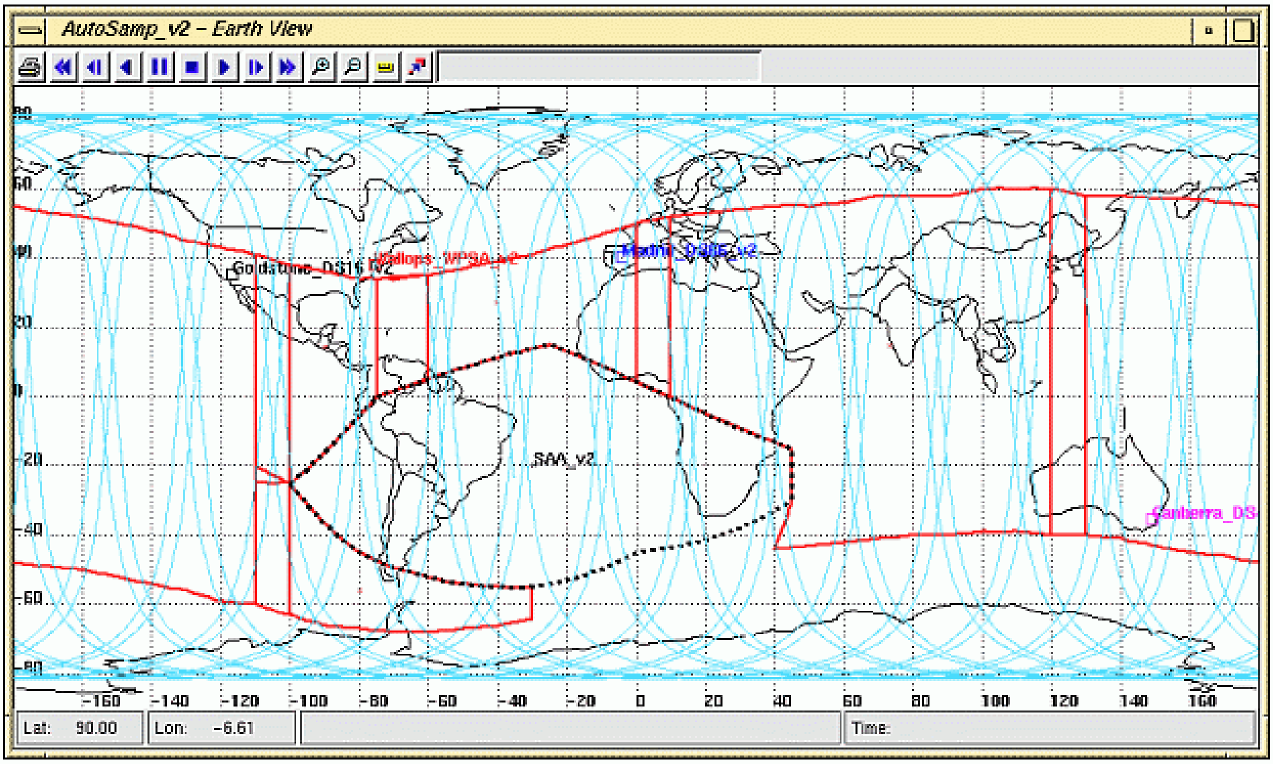


Figure 3: STK map window showing the South Atlantic Anomaly as well as the area targets defined for the SAMPEX mission

Accurate Orbit Determination Made Easy

To aid in the studies and measurements being conducted by the SAMPEX satellite, IMACCS was also designed to perform predictive orbit analyses. CSC chose add-on modules to accurately predict and propagate SAMPEX's orbit. Using measurements of SAMPEX's location over a period of time, CSC used a separate module to precisely determine the current orbit of the satellite. The module was then used to propagate the satellite's orbit into the near future. This information was then provided to the control center for mission planning and scheduling purposes.

CSC utilized another add-on module to simplify the process of scheduling and categorizing SAMPEX data, which allowed the user to link and/or group objects together—a powerful capability that enables the user to determine multiple accesses or multi-hop accesses. The user can create a constellation, which groups like objects together and establishes access constraints for the group of objects. Connectivity problems between multiple ground- and space-based objects or sets of objects can be rapidly analyzed with this feature set.

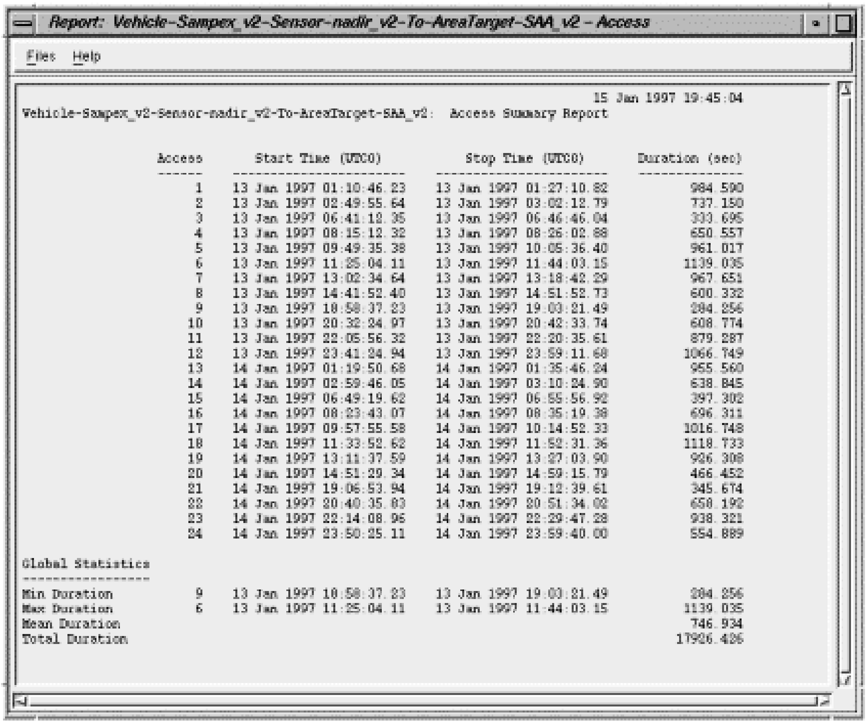
For the SAMPEX mission, CSC engineers were able to use the COTS package to determine station contact times for multiple ground stations simultaneously. The software was not limited to the number of ground stations or the location of the ground stations; the operators could even use Boolean operators to define connectivity requirements and priorities. Using this contact information, CSC could subsequently determine data collection times for each of the specific regions with unparalleled ease.

Generating Reports

IMACCS used several built-in report styles to document analysis results and orbital parameters. Reports documented the lighting times for the satellite as well as a variety of access times from the satellite to each facility, to all of the facilities (defined as a constellation), and to the South Atlantic anomaly (defined as an area target). In addition, reports summarizing azimuth, elevation and range data from the satellite to various facilities were generated.

The reports were then provided to the control centers as well as satellite schedulers for planning purposes.

One of the primary benefits of a COTS system is the fact that the technology has already been developed. The IMACCS prototype was developed in only 90 days. It was implemented on an operational satellite and successfully demonstrated in a short period of time. COTS products are typically more flexible than those developed in-house on a mainframe or workstation. In addition, commercial tools are cross-platform compatible so that changing hardware requirements do not impact software performance. Most of the COTS software used in IMAACS runs on most UNIX platforms and is also available on the PC. STK data is saved in ASCII format for cross-platform compatibility and for easy export, if required.



Report: Vehicle-Sampex_v2-Sensor-nadir_v2-To-AreaTarget-SAA_v2 - Access

Files Help

15 Jan 1997 19:45:04

Vehicle-Sampex_v2-Sensor-nadir_v2-To-AreaTarget-SAA_v2: Access Summary Report

Access	Start Time (UTCG)	Stop Time (UTCG)	Duration (sec)
1	13 Jan 1997 01:10:46.23	13 Jan 1997 01:27:10.82	984.590
2	13 Jan 1997 02:49:55.64	13 Jan 1997 03:02:12.79	737.150
3	13 Jan 1997 06:41:12.35	13 Jan 1997 06:46:46.04	333.695
4	13 Jan 1997 08:15:12.32	13 Jan 1997 08:26:02.88	650.557
5	13 Jan 1997 09:49:35.38	13 Jan 1997 10:05:36.40	961.017
6	13 Jan 1997 11:25:04.11	13 Jan 1997 11:44:03.15	1139.035
7	13 Jan 1997 13:02:34.64	13 Jan 1997 13:18:42.29	967.651
8	13 Jan 1997 14:41:52.40	13 Jan 1997 14:51:52.73	600.332
9	13 Jan 1997 18:58:37.23	13 Jan 1997 19:03:21.49	284.256
10	13 Jan 1997 20:32:24.97	13 Jan 1997 20:42:33.74	608.774
11	13 Jan 1997 22:05:56.32	13 Jan 1997 22:20:35.61	879.287
12	13 Jan 1997 23:41:24.94	13 Jan 1997 23:59:11.68	1066.749
13	14 Jan 1997 01:19:50.68	14 Jan 1997 01:35:46.24	955.560
14	14 Jan 1997 02:59:46.05	14 Jan 1997 03:10:24.90	638.845
15	14 Jan 1997 06:49:19.62	14 Jan 1997 06:55:56.92	397.302
16	14 Jan 1997 08:23:43.07	14 Jan 1997 08:35:19.38	696.311
17	14 Jan 1997 09:57:55.58	14 Jan 1997 10:14:52.33	1016.748
18	14 Jan 1997 11:33:52.62	14 Jan 1997 11:52:31.36	1118.733
19	14 Jan 1997 13:11:37.59	14 Jan 1997 13:27:03.90	926.308
20	14 Jan 1997 14:51:29.34	14 Jan 1997 14:59:15.79	466.452
21	14 Jan 1997 19:06:53.94	14 Jan 1997 19:12:39.61	345.674
22	14 Jan 1997 20:40:35.83	14 Jan 1997 20:51:34.02	658.192
23	14 Jan 1997 22:14:08.96	14 Jan 1997 22:29:47.28	938.321
24	14 Jan 1997 23:50:25.11	14 Jan 1997 23:59:40.00	554.889

Global Statistics

Min Duration	9	13 Jan 1997 18:58:37.23	13 Jan 1997 19:03:21.49	284.256
Max Duration	6	13 Jan 1997 11:25:04.11	13 Jan 1997 11:44:03.15	1139.035
Mean Duration				746.934
Total Duration				17926.426

Figure 4: Sample report summarizing South Atlantic Anomaly "fly-through" times for SAMPEX

Far Ultraviolet Spectroscopic Experiment (FUSE)

As a value-added supplier of commercial-off-the-shelf (COTS)-based command and control solutions, Interface & Control Systems, Inc. (ICS) has placed great emphasis on utilizing commercial products for satisfying specific, well-defined system requirements. ICS is under contract to the Johns Hopkins University as a team member of the FUSE Program. The FUSE satellite, a medium explorer mission, is designed to study the origin and evolution of the lightest elements – hydrogen and deuterium, and the forces and processes involved in the evolution of galaxies, stars, and planetary systems. ICS is responsible for the day-to-day management of the instrument and spacecraft activities within the Satellite Control Center, of which two integrated COTS products – SCL and STK– play an integral role. ICS realized that by using COTS products, their team would be guaranteed continued support for the entire mission. At a minimum the software met or exceeded the initial set of requirements, plus the team benefited from routine upgrades and the software's expandability and ability to interact with other products in the integrated software environment. COTS is used extensively for the operations of the FUSE satellite including:

- Real-time telemetry processing for on-orbit position and attitude visualization
- Generation of contact times
- Orbit propagation
- Generation of pointing angles for telescope operations
- Star tracking

The FUSE program was delivered at a reduced mission cost of $ 108 million from the original budget of $ 254 million, and the satellite was launched 2 years earlier than originally planned.

The Future Is Now – Ellipso™

The Ellipso satellite system is a constellation of communications satellites developed by Mobile Communications Holding, Inc. The Ellipso constellation is comprised of two subconstellations: Concordia and Borealis. The Concordia constellation consists of six operational satellites in circular equatorial orbits with an altitude of 8068 km. The satellites in the Concordia constellation provide Earth ground coverage between the 55° North and 55° South parallels.

The second subconstellation, Borealis, consists of five operational satellites in each of two orbital planes. Both orbital planes are critically inclined and sun-synchronous. The Borealis constellation will not utilize the typical critical inclination of 63.4° s but will instead use the complementary retrograde inclination of 116.6° . Satellites in the first of the Borealis orbit planes have a local time of ascending node of noon, while those in the second orbit plane have a local time of ascending node of midnight. The Borealis™ orbits are eccentric with a perigee of 520 km and an apogee of 7846 km, where perigee remains in the southern hemisphere. The satellites in the Borealis constellation provide additional coverage over the northern hemisphere.

The Ellipso satellite system, sometimes called Ellipsat, is the result of recent advances in communications technology with a highly innovative approach to using satellites. Its goal is to provide a simple, pragmatic and economical near-global telecommunications system. The FCC granted Ellipso a license on June 30, 1997. Once operational, it will provide combined position determination and mobile voice services using up to 17 satellites. Because of its unusual orbits, the Ellipso configuration is able to provide coverage to a large area with a small number of satellites.

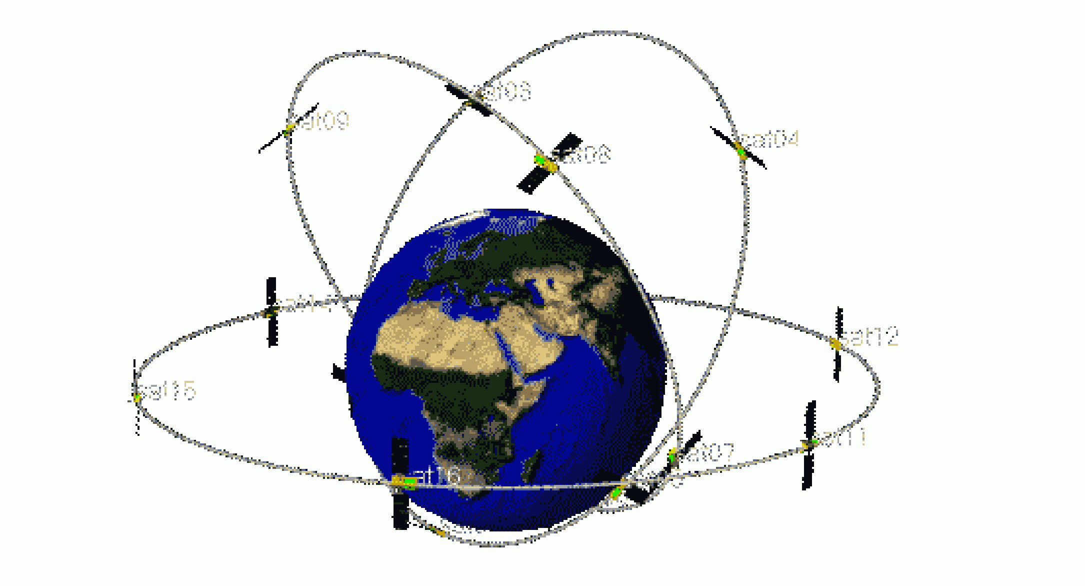


Figure 5: Graphical representation of the two constellations comprising the Ellipso satellite system

Spectrum Astro, Inc. performed analysis to determine the orbital stability and propellant budgets for the Ellipso system. To ensure that the launch itself is accomplished in the most economical manner while providing optimal satellite positioning, Mobile Communications Holding, Inc. contracted with Spectrum Astro to provide mission planning analyses. Spectrum Astro began designing the satellite and studying the constellation configuration as part of the Ellipso's conceptual orbit design phase.

Since the coverage characteristics of the system depended on the relative positions of the satellites and orbital planes, the effects of various orbit perturbations such as atmospheric drag, third body gravity, solar radiation pressure, and gravitational resonance on each subconstellation had to be well understood. A thorough understanding of the perturbations was especially important for the Borealis constellation since the planned time frame for launch during the year 2000 falls during a maximum in the solar activity cycle. The effect of atmospheric drag in the region near perigee of the Borealis orbits will be significantly increased due to the higher level of solar activity. As a result, more drag make-up maneuvers would be needed than would be necessary during times of lower solar activity.

The satellites that comprise the Ellipso system are designed to remain operational over a 7-year period. During this time, the satellites must maintain their orbits despite atmospheric drag at perigee. Spectrum Astro conducted an analysis of the long-term orbit evolution of the satellites. Because the Ellipso orbits are highly refined in shape and orientation, Spectrum Astro used STK and a number of add-on modules to model the satellites and their orbital planes. One of the goals of the Ellipso satellite system is to keep the orbits stable enough so that they pass over the same spot, at approximately the same time every day. Using commercial software, analysts are able to perform precise orbit predictions over the 7-year time period. Access reports are reviewed to identify orbit drift trends. Based on these trends, analysts can determine the best time for fuel burns to correct orbit drifts. Results enabled Spectrum Astro to determine the best way to maintain vehicle shape and orientation over the 7-year period by burning the least amount of fuel every few orbits.

This analysis phase is especially important for the Ellipso satellites because there is no historical data for this type of orbit. The need to maintain the constellation geometry over this time span means that the determination of accurate propellant budgets is critical. The goal of this effort is to determine a common sizing for the propulsion system of all satellites in the Ellipso constellation.

Command & Control. The Ellipso satellite will be maintained using three ground command and control stations. Personnel at these stations will determine how often fuel burns should be performed to keep the satellites in orbit. They will also analyze and track the contact times between the satellites and the ground stations. To successfully fulfill their missions, the locations for the ground stations must be carefully chosen. In addition, the satellites themselves must have antennas that are positioned for optimal access.

To ensure that communications between satellites and ground stations are successful, Spectrum Astro conducted analyses to determine the number and position of antennas on the satellites as well as the optimal location for the three C&C ground stations. The Ellipso satellite system featured 61 different communications beams. To better define communications coverage, Spectrum Astro studied how often these beams overlap, which beams are used heavily, and how much coverage is provided on average.

In addition to mission planning and maintenance, analysts must also provide a plan for deorbiting, or graveyarding, the satellites at the end of their useful life. For the Concordia constellation, the orbital altitudes will most probably be raised so that the next set of satellites can be launched into orbit directly beneath the first set. Due to the elliptical nature of the Borealis constellation, operators will most likely perform a graveyard burn so that the satellites reenter the Earth's atmosphere in a few years. Graveyarding itself is dependent on the amount of fuel required for deorbiting and analysis tools such as STK can be used to determine reasonable opportunities for the deorbit of the satellites.

For the Ellipso program, Spectrum Astro used COTS software to perform mission analysis tasks. Spectrum Astro considers flexibility a valuable feature of COTS software. Analysts are able to export data for mission-unique analysis using other software packages that were developed in-house. They also import orbit trajectory data generated by mission-specific software into the COTS software for analysis. This flexibility allows Spectrum Astro to exploit all options for analysis without limiting the format and structure.

Spacecraft Anomaly Resolution and Rescue

The complexity of these space systems continues to increase as the demand for satellite services of various functionality expands to global proportions. The dependence of the success of these satellites on reliable hardware and software systems also increases proportionally, as does the corresponding risk of failure of these systems. In the last two years alone, nine spacecraft experienced hardware or software malfunctions which rendered them temporarily unusable but recoverable. As short-term hardware replacement on in-orbit spacecraft is impossible, a flexible software package designed to address and resolve all types of spacecraft anomalies is the ideal means by which to recover the spacecraft.

COTS software packages have the key advantages of flexibility and breadth of functionality to achieve anomaly resolution. Traditional, in-house software packages are generally developed to address normal operations of a particular satellite or system. Due to the budgetary concerns of developing a complete software system specific to a spacecraft or system, satellite companies often do not have the means to incorporate anomaly-resolution tools into their in-house software. COTS systems, which offer the benefit of lower cost, go beyond just meeting the normal operational requirements during nominal space-systems operation. Flexibility and breadth of functionality are vital due to the many types of anomalies which can be encountered and the procedures required for resolution, including:

- Accessing the Space Surveillance Network (SSN)
- Orbit determination
- Orbital lifetime remaining
- Orbit maneuver
- Attitude determination
- Attitude maneuver
- Collision avoidance
- Solar illumination analysis (Power and Thermal)
- Determination of communication availability
- Determination of ground station availability
- Mission capability analysis

Validated and verified COTS systems that have been proven on many satellite missions, those, which address many types of anomalies encountered and perform the functions required for resolution, are the ideal tools for these tasks. For the relatively small investment required, typically tens of thousands of dollars, these packages can be, at a minimum, an inexpensive insurance policy. In several cases throughout the past few years, however, COTS packages have been used to rescue billion-dollar satellites. The fast, flexible analysis these software packages afford have already helped save billions of dollars of space systems and will undoubtedly continue to be a vital tool to current and new systems as the proliferation of satellites continues.

References

1. SpaceVest and KPMG Peat Marwick, *State of the Space Industry* (KPMG, 1997).
2. Stephen A. Book, *Cost-Risk Analysis Seminar – Its Role in Proposal Preparation, System Development, and Program Control* (The Aerospace Corporation, 1999).

PAUL L. GRAZIANI is the President and Chief Executive Officer of Analytical Graphics, Inc. (AGI), the producer of the Satellite Tool Kit (STK)® software suite, leading commercial-off-the-shelf software package for the space industry. Mr. Graziani can be reached via phone: +1 610-578-1000; fax: +1 610-578-1001; or

e-mail: paul@stk.com. The AGI web site is located at http://www.stk.com.

Satellite Tool Kit is marketed internationally by International Aerospace that can be contacted via phone +44 1908 551105, Fax +44 1908 551063 or e-mail: info@intaero.com.

BACK TO TOP

Information & Security. An International Journal
e-mail: infosec@mbox.digsys.bg

Commercial-Off-The-Shelf Software Becomes Mission-Critical to Success and Cost-Effective Space Missions

Paul L. Graziani

Keywords: COTS software, satellite, simulation, space surveillance network, space-based communications

Abstract: Many organizations can no longer afford to develop mission-specific software in-house. Commercial satellite software that can be used for multiple missions is playing a growing, cost-saving role in enabling not only successful but economical space activities. Commercial off-the-shelf (COTS) software that is stable, accurate, and well-supported is key in all phases of a satellite's life—from pre-launch analyses to ground support through to graveyarding The paper shows the benefit of using COTS software in the big international projects, related to construction of reusable, generic software "building blocks" based on legacy software. In theory, new missions could then select among these building blocks to build systems in an object-oriented manner.

Authors: **Mikel D. Petty, Robert W. Franceschini, Amar Mukherjee**
Title: **A Terrain Reasoning Algorithm for Defending a Fire Zone**
Year of issuance: **1999**
Issue: **Information & Security. Volume 3, 1999**
Hard copy: **ISSN 1311-1493**

A TERRAIN REASONING ALGORITHM FOR DEFENDING A FIRE ZONE

Mikel D. PETTY, Robert W. FRANCESCHINI and Amar MUKHERJEE

Table Of Contents:

---

1. Background

This section provides brief background information on distributed simulation, computer generated forces, and ModSAF, which set the context for the research described in this paper.

1.1. Distributed simulation

Distributed simulation is an approach to building large-scale simulation models from a set of independent simulator nodes communicating via a network. The simulator nodes each independently simulate the activities of one or more entities in the simulated world and report the attributes (e.g., location and velocity) and actions (e.g., weapons firing) of those entities via the network to the other simulator nodes using a pre-defined communications protocol. In a typical distributed simulation, the simulated entities coexist in a common simulated environment and can interact with each other; their interactions are realized via the exchange of messages in the protocol.

One example of a military distributed simulation is the U.S. Army's Close Combat Tactical Trainer (CCTT). In CCTT, which is a real-time system, the simulator nodes typically represent single vehicles, such as tanks or aircraft. CCTT simulators are substantial devices consisting of a simulation computer, a computer image generator, and a physical simulation of the interior of the vehicle being simulated. The human crew of the simulator maneuver their simulated vehicle in the simulated world and interact with (e.g., fire their weapons at) other vehicles.

During the execution of a simulation, each simulator's simulation computer tracks the position of the vehicle it is simulating, as well as the other entities in the simulation, on a terrain database common to the system. The simulator broadcasts messages reporting the location of the vehicle it is simulating, which are used by the other simulators to track that location, and may send other messages to mediate inter-entity interactions. The specific network protocol used by the CCTT simulators is a version of the Distributed Interactive Simulation (DIS) IEEE-1278 standard.

1.2. Computer generated forces

Distributed simulations often include a special type of simulator node known as a Computer Generated Forces (CGF) system. A CGF system usually generates multiple battlefield entities (such as tanks, aircraft, and infantry) using computer algorithms rather than human crew to generate and control the real-time actions of those entities. The behavior algorithms are designed to react to the simulation situation and to generate tactically and doctrinally correct actions. CGF systems are connected into the distributed simulation in the same manner as the other nodes and send and receive the requisite network messages for their CGF entities. CGF systems are often used to provide hostile forces and supplementary friendly forces in battlefield training simulations that involve human trainees. CGF systems may also be used for analytical simulations, where they often populate the battlefield completely as needed for the analytical experiment.[1]

1.3. ModSAF

One CGF system of particular interest in distributed simulation is ModSAF (Modular Semi-Automated Forces). ModSAF is the most widely distributed CGF system and is used for many purposes, including training, analytical experiments, and research into CGF algorithms. (However, ModSAF is not used in CCTT, which has its own CGF system). It is a large system, consisting of approximately 1.5 million lines of C code. It includes user interface, network interface, physical modeling, low (entity) level behavior generation, and high (military unit) level behavior generation capabilities.[2]

ModSAF represents the terrain of the battlefield using a data format known as CTDB (Compact Terrain Database Base). In the CTDB format, the terrain surface is primarily represented by elevation posts, which are elevation z values for the terrain at a regular square grid of x, y locations. Triangles are induced from the elevation posts and these triangles make up the terrain surface. Terrain features, such as roads, bridges, buildings, individual trees, tree lines, and tree canopies are also represented in the CTDB format.[3]

2. Introduction to the research

This section defines the problem, briefly mentions some related research, and identifies existing algorithms to solve the problem.

2.1. Problem definition

When a military unit (platoon, company, or battalion) takes up a defensive position, finding good deployment locations for the individual entities of that unit is an important terrain reasoning task. This is true for both real and simulated military units. A military unit that is being generated by a CGF system will consist of some number of entities (typically 3 to 50). A useful CGF capability is to be able to order a unit to take up a defensive position and have a terrain reasoning algorithm determine deployment locations for each of the individual entities of that unit. The CGF system then moves the entities to the selected locations. Such an algorithm in fact exists in the ModSAF CGF system, in a behavior called *Hasty Occupy Position*. However, the Hasty Occupy Position algorithm does not consider observation of the area to be defended. We sought improved performance with an algorithm that employs geometric terrain reasoning to consider observation.

We will refer to the military unit for which deployment locations must be chosen as the *unit*. Its component units at the next lower echelon will be called *subunits*. In this work, we confined our attention to company units and platoon subunits.

The process of selecting specific locations for the individual entities of the unit must begin with certain given information. The *battle position* for a unit is a specification of the region in which its entities must be located; it is given as a chain of segments located on the terrain. The length of the battle position, in both military tactics and in simulation, depends on the size of the unit and the tactical situation. A typical battle position for a company is 1500 meters long.[4] The entities of the company deploy within some radius of the battle position segments, where that radius is another piece of given information. We will refer to points within the given radius of the battle position chain of segments as *in* or *within* the battle position. Also given is the *engagement area*, which is a region of terrain through which an enemy attack is expected. (The expected direction of attack is determined by reconnaissance and military intelligence and the engagement area is determined from that direction by command planning at echelons above the unit. For this problem, the engagement area is simply given.) An engagement area for a company is typically 1000 to 1500 meters wide and 2000 to 3000 meters deep. The entities of the unit should be located where they can observe as much of the engagement area as possible in order to employ direct fire against attacking entities. A *location* is a specific point on a terrain database. *Target reference points* (TRPs) are locations that are used to delimit and partition the engagement area for the subunits of the unit.

Cover is protection from enemy fire and observation by intervening terrain surface and *concealment* is protection by terrain features. Cover or concealment may only partially protect an entity. For example, a tank in hull defilade might be located so that its hull is covered by a ridge crest but its turret is not, allowing it some protection while still allowing it to use its weapons, a desirable situation.

To select deployment locations for the entities of a unit three types of constraints must be considered. First, the geometry of the terrain surface and the terrain features and their effect on intervisibility obviously constrain how much of the engagement area can be observed from any location in the battle position. The goal is to observe the engagement area; so that forces "... must see the whole battlefield ...".[5]

Second, cover and concealment in the battle position must be considered, as the defending entities should not be unduly exposed to enemy fire. Army doctrine for the defense states that "... fighting forces must be covered and concealed ..." and "Tanks engage enemy [forces] from covered and concealed positions."[6] For the fire zone defense problem, the entities should be deployed in cover and concealment locations where the entity is partially but not fully covered or concealed as the entity must be able to use its weapons to defend the engagement area.

Finally, military doctrine imposes constraints on where the entities may be located. Two are considered here: first, a unit battle position should be partitioned into battle positions for the subunits. According to 1993 U.S.Army Field Manual,[7] in defensive operations "Commanders position their forces in platoon, company, or battalion battle positions ...". Second, to retain tactical cohesiveness the entities of a subunit should be kept together rather than intermingled.[8]

With all these concepts in hand we can state the *fire zone defense* problem:

> Given a military unit, a terrain database, and a battle position, engagement area, and target reference located on that database, find deployment locations within the battle position for the entities of the unit that simultaneously satisfy the intervisibility, cover and concealment, and doctrinal constraints and also maximize observation of the engagement area.

To evaluate solutions to the problem as stated, we need a performance metric that is both militarily meaningful and quantitatively comparable. To define such a metric we turn to military sources, which make the goal of selecting deployment locations very clear. "Weapons are most effective when sited to cover an expanse of terrain over which the attack must come."[9] As stated in the U.S. Marine Corps doctrine, "Weapons must be located where their effects will be greatest and dead space is minimized."[10] ("Dead space" is terrain that can not be observed or fired on from a battle position because of intervening terrain).

In the context of automated terrain reasoning, intervisibility measures for points are suggested by Keirsey and coworkers.[11] The first, *general visibility*, is the cumulative area of terrain visible from a point. The second, *specific visibility*, is the percentage of a specified area visible from that point. The second measure is most relevant in this problem, where the specified area is the engagement area. Therefore, based on the referenced sources, we define *location observation* as the portion of the engagement area visible from a location and *cumulative observation* as the sum of the location observation values for all the deployment locations selected for the entities of a unit. Algorithms to solve the fire zone defense problem will be compared using the cumulative observation metric.

2.2. Related research

A 2-dimensional algorithm that positions a set of sensors within a set of territories so as to optimally cover the given territories with the envelopes of the sensors is presented by.[12] A symbolic object-oriented method for finding positions for a unit's component subunits (e.g. the platoons of a company); is described by Hieb, Hille and coworkers [13]; the method considers the terrain as a set of symbolic conceptual objects (e.g. hill-863 or avenue-of-approach-2) and consequently does not produce precise deployment locations. In work reported by Rajput and Tu, [14] the ModSAF cover and concealment finding algorithm was enhanced to find assault positions and support-by-fire positions, which are terrain areas used by units when attacking an enemy position.

2.3. Existing algorithm

ModSAF version 2.1 was used for this experiment, as it was the version available to the authors when this work was performed. The ModSAF algorithms described herein, and in general all references to ModSAF in this paper are with respect to that version. More recent versions of ModSAF exist and may differ from these descriptions.

To evaluate the effectiveness of our new algorithm for the fire zone defense we compared it with ModSAF's Hasty Occupy Position (HOP) algorithm. The HOP algorithm is described by Courtemanche and Ceranowicz [15] and in the ModSAF documentation. In the former it is referred to as "Hasty Occupy Battle Position" but it is "Hasty Occupy Position" in the ModSAF implementation and operator interface; we will use the latter term. The HOP algorithm controls the behavior of ModSAF units at both the company and platoon levels. The company HOP algorithm partitions a company battle position, supplied by the operator as a chain of segments, evenly into platoon battle positions and passes them to the platoon HOP algorithm. The platoon HOP algorithm calculates a battle area, which is a rectangular area centered on the battle position, for each entity in the platoon. It then finds a deployment location for the entity within its battle area using the ModSAF cover and concealment algorithm. If no covered or concealed location is found within the battle area, the entity is placed at the center of the battle area on the battle position segment; these are the HOP default locations.

The cover and concealment algorithm finds covered and concealed locations on the terrain relative to a given location. It does so by extending lines from the given location through the battle area and examining the terrain profile along those lines, using a spacing parameter.[16]

3. Fire Zone Defense algorithm

Overview. The ModSAF HOP algorithm was replaced with a new algorithm, called Fire Zone Defense (FZD), to solve the stated problem of selecting locations for the entities of a unit so as to effectively defend an engagement area. More precisely, specific parts of the HOP algorithm were replaced. This section describes the FZD algorithm and its integration into ModSAF.

Input. The FZD algorithm is invoked from the ModSAF operator interface as a "Hasty Occupy Position" order. The operator provides via the operator interface the battle position (a chain of segments), left and right TRPs for the engagement area (points), and the engagement area itself (a quadrilateral), and assigns a unit to defend that engagement area. As implemented for this work, the FZD algorithm expects a company unit. The algorithm uses that explicit input and the terrain database to find the deployment locations for the entities of the unit.

Locations. A *location* has already been defined as a specific point on the terrain surface. Certain categories of locations are central in the FZD algorithm. *Sighting locations* are locations within an engagement area that should be visible from the battle position so as to effectively observe the engagement area. Ideally, we might want to sight every location in the engagement area, but because there are an infinite number of such locations, it is impractical to consider all locations in the engagement area when evaluating observation. Sighting locations are a subset of the locations in the engagement area chosen to be representative of the entire engagement area, so that observing the sighting locations is equivalent to observing the entire engagement area, to a good approximation. Sighting locations are calculated by the FZD algorithm using simple geometric terrain analysis. In particular, sighting locations are generated for two types of terrain locations. First, the terrain surface at the elevation posts are taken as sighting locations. Recalling that the CTDB is primarily an elevation post format, and that the elevation posts induce triangles (which are of course planar), the idea is that if the elevation post locations can be observed then the triangles between them can be as well. Hence elevation post locations within the engagement area serve as sighting locations. In the CTDB format the elevations posts are spaced on a 125 meter grid, so the number of elevation posts in a typical engagement area is reasonable. Second, to account for the effect on intervisibility of terrain features, the vertices of terrain features within the engagement area are also taken as sighting locations. Buildings, tree lines, and canopies are included, but individual trees are not. Van Brackle and coworkers [17] introduced in the reconnaissance planning work the notion of sighting locations representing a terrain area.

Battle position locations are locations evenly spaced along the segments of a battle position. The spacing is a parameter to the algorithm. Battle position locations are used to partition a unit battle position into subunit battle positions, as will be seen later. *Defensive locations* are locations within the battle position that are covered or concealed relative to the engagement area, or a TRP within it. Defensive locations are found using the ModSAF cover and concealment algorithm by passing to it a rectangle and TRP reference point calculated from the battle position. *Deployment locations* are defensive locations actually selected by the FZD algorithm as the optimal locations for the defending unit's entities. Note that the set of deployment locations is a (not necessarily proper) subset of the set of defensive locations.

Observation value. The FZD algorithm depends on the notion of observation value for a location. The observation value for a location is an approximation of the quantity previously defined as location observation, which was how much of the engagement area is visible from a location, with two added factors. First, observation value also considers range. Only those sighting locations within a given range of the given location are considered in the observation value. Second, a weight with values between 0.0 and 1.0 associated with each sighting location is included. The weight is a measure of how important it is to observe that sighting location. Initially all sighting locations have weights of 1.0. We shall see how those weights are changed later.

The FZD algorithm computes the observation value for a given location by invoking the ModSAF point to point intervisibility routine, which returns a value between 0 (not visible) and 1 (completely visible). The observation value for a given location is the sum, for every sighting location in the engagement area within a given range of the given location, of the product of the intervisibility value between the given location and the sighting location and the sighting location's weight. Formally, the observation value for location d and range r is defined as

$$\sum_{s \in S} w(s) \times v(d,s) \times g(d,s,r)$$

where d is the given location, r is the given range, S is the set of sighting locations, $w(s)$ is the weight of sighting location s, $v(d,s)$ is the ModSAF intervisibility value from d to s, and $g(d,s,r)$ is a function that returns 1 if sighting location s is within range r of d and 0 otherwise.

Given a set D of defensive locations and a set S of sighting locations, we define the *intervisibility* graph as a complete bipartite graph $V = (D \cup S, E)$ where the edges in E connect every defensive location $d \in D$ to every sighting location $s \in S$. For each edge $(d,s) \in E$, the ModSAF intervisibility value $v(d,s)$ is stored in an array, which represents the graph. Each array entry also stores the distance between d and s, denoted $dis(d,s)$. Once the intervisibility and distance values in the array have been filled in, the observation value for a defensive location $d \in D$ can be computed from the array without invoking the point to point intervisibility routine or performing a distance computation. This saves execution time if the observation value for the locations in a known set will be needed repeatedly, as is the case for defensive locations in the FZD algorithm.

Sighting location weights. As mentioned, associated with each sighting location is a weight with values between 0.0 and 1.0. Initially all sighting locations have weights of 1.0. The weight measures how important it is to observe that sighting location. From the viewpoint of tactical effectiveness, the importance of observing a particular sighting location by an entity declines as a function of how many other entities can also observe that sighting location. In other words, for each entity that can observe a sighting location, it becomes relatively less important to deploy another entity to observe that same sighting location as compared to sighting locations observed by fewer or no entities. The idea is that important for each entity to be able to see as much of the engagement area as possible, but it is also important for all or most locations in the engagement area to be observed by some entity.[18] The FZD algorithm reflects this military consideration. It sequentially chooses deployment locations from among the defensive locations for the entities of a unit, and as each one is chosen, the weights of the sighting locations are reduced. Specifically, for a selected defensive location d, and for every sighting location s Î S, $w(s)$ is reduced to

$$w(s) - (h \times v(d,s) \times w(s))$$

The reduction factor h is an algorithm parameter; it was set $h = 0.5$ for this task. Different values would affect the dispersal of the selected deployment locations. The weight reduction calculation depends on the intervisibility value $v(d,s)$ between d and s, so if s is not highly visible from d, its weight is not reduced much.

Algorithm. For a formal statement of the FZD algorithm, see [19]. Informally, it consists of these steps:

Steps 1 through 3 are executed once for the company.

1. Generate a set of sighting locations in the engagement area. This is done by accessing the CTDB, retrieving all elevation posts and feature vertices within a minimal square enclosing the engagement area, and testing each for enclosure within the engagement area.

2. Partition the company battle position into platoon battle positions and compute platoon TRPs.

    2.1 Generate battle position locations evenly spaced along the battle position segments. Compute, store, and accumulate the observation value for each battle position.

    2.2 Divide the total of the battle position location observation values by the number of subunits to get the subunit share of the total observation value.

    2.3 Step along the battle position locations summing the battle position location observation values. When the accumulated sum exceeds a subunit share, that battle position location is the end of one subunit battle position and the beginning of the next. In this way the unit battle position is partitioned into subunit battle positions at the battle position locations so that the sum of the observation values for the battle positions within each subunit battle position is approximately equal to the subunit share computed in step 2.2. This has the effect that portions of the unit battle position which provide relatively poor observation of the engagement area are included in larger subunit battle positions and portions with relatively good observation become smaller subunit battle positions, concentrating more entities into portions with good visibility. The subunit battle positions, like the unit battle positions, consist of chains of one or more segments; some of the segments in the subunit battle positions may be subsegments of segments in the unit battle position. Figure 1 gives an example of this process.

    2.4 Compute the subunit TRPs by dividing an arc between the unit TRPs in proportion to the division of the unit battle position computed in step 2.3. (The arc is a subset of the circle centered on the center of the battle position and passing through the unit TRPs.)

3. Invoke the platoon FZD behavior for each platoon of the company, passing to each one a subunit battle

position and the left and right TRP pair computed in step 2 as well as an engagement area TRP computed as the centroid of the engagement area polygon.

Steps 4 through 8 are repeated for each platoon.

4. Construct a platoon battle area rectangle on the platoon battle position with its width extending from one endpoint of the platoon battle position to the other and its depth given by a ModSAF parameter.

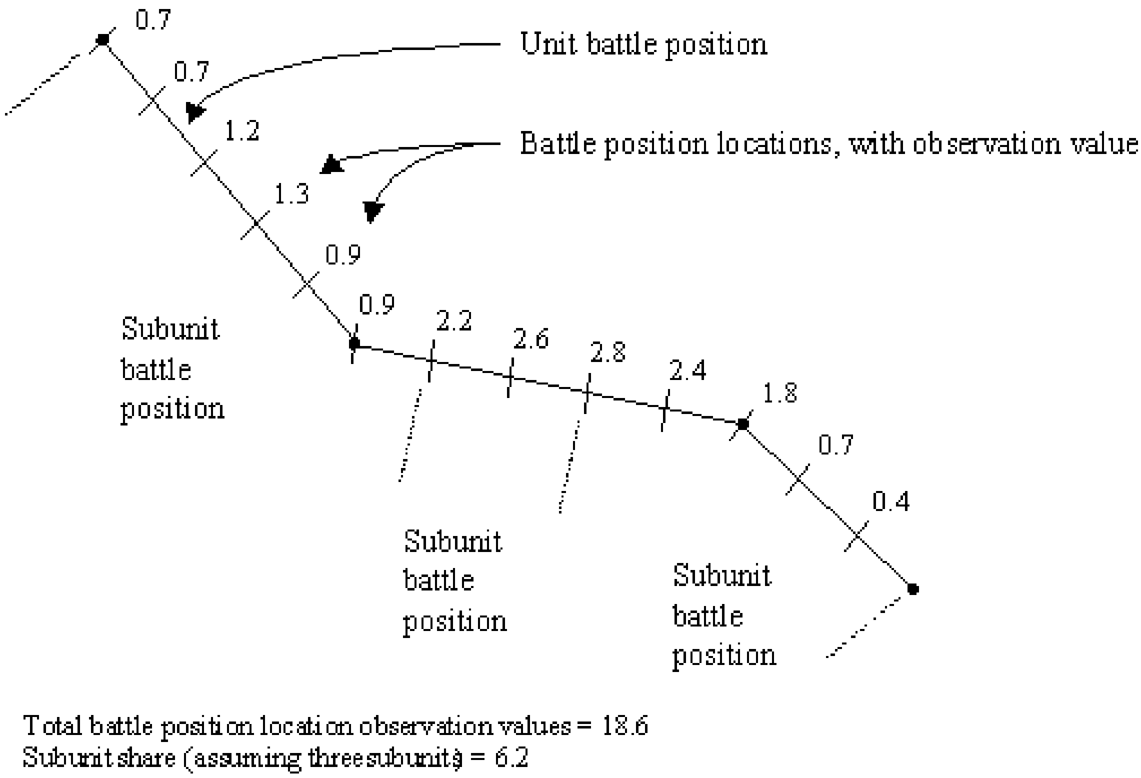


Figure 1. Unit battle position partitioning example.

5. Generate a set of defensive locations for the entities of a platoon by invoking the ModSAF cover and concealment algorithm, passing to it the platoon battle position rectangle and the engagement area TRP. This produces a set of defensive locations for the platoon which are the covered or concealed locations within its battle area rectangle.

6. Fill in the array representing the intervisibility graph *V* by determining the point to point intervisibility value $v(d,s)$ and distance $dis(d,s)$ between each defensive location and each sighting location.

7. Select one of the defensive locations as a deployment location for each entity of the subunit using a greedy optimization procedure.
For each entity:

7.1 For each unselected defensive location *d*, determine the observation value of that location using the intervisibility graph array and the sighting range of the entity. Save the defensive location with the best observation value.

7.2 Select the defensive location with the best observation value; assign it to the entity as its deployment location and mark it as selected.

7.3 Reduce the weights of the sighting locations for the selected defensive location.

8. Invoke the ModSAF route planning and movement functions to move the individual entities of the subunit to their assigned deployment locations.

Optimization comments. The fire zone defense algorithm selects a subset of the defensive locations for the entities. In doing so it attempts to maximize total observation value from the selected defensive locations, i.e., to maximally include the sighting locations in areas that are visible and within range of the selected defensive locations. It does so using a greedy approach, selecting for each entity the defensive location not already assigned to an entity that has the largest observation value.

A strict mathematical maximization of sighting location inclusion that does not consider range and inclusion by previous entities would not necessarily be tactically optimum. The algorithm must balance competing goals such as locating entities where they can observe the largest number of sighting locations against locating entities so as to include every point in the set of sighting locations within a militarily sufficient number of fields of fire. The combination of a greedy selection with a reduction of the sighting location weights after each defensive location selection is a heuristic intended to achieve good tactical effectiveness.

4. Evaluation experiment

This section reports the experiment conducted to compare the HOP and FZD algorithms.

4.1. Experiment overview

The HOP and FZD algorithms both perform the same task: given a CGF unit ordered to take up a defensive position relative to an engagement area, determine good deployment locations for each of the individual entities of that unit. The obvious experiment to compare the two algorithms is to have both perform that task under identical circumstances and evaluate their performance. That obvious approach is precisely what was done. A set of trials, each consisting of a terrain locale, a unit battle position, left and right TRPs, and an engagement area, were created. The HOP and FZD algorithms were run for each trial to produce a deployment for the entities of the unit. Each deployment was evaluated in terms of the cumulative observation metric and the results were compared.

4.2. Experiment design

Trial characteristics. The HOP and FZD algorithms were compared over a sequence of trials. Each trial consisted of a given terrain database, locale on that terrain database, battle position, engagement area, left and right unit TRPs, and military unit. Two runs were conducted for each trial, one using the HOP algorithm and one using the FZD algorithm.

In order to provide comparable values and eliminate confounding variables, most of the characteristics of the experiment remained fixed over all trials and runs. In particular, these items did not change:

1. Terrain database (the Ft. Knox CTDB version 5)
2. HOP and FZD algorithm implementations
3. ModSAF cover and concealment algorithm
4. Cover and concealment spacing (5.0 meters)
5. Military unit (U.S. M1 tank company)

Certain items varied with the trial, but were fixed for the two runs (HOP and FZD) of a trial. Those were:

1. Terrain locale
2. Engagement area

3. Left and right unit TRPs
4. Unit battle position

Of course, the algorithm used (HOP or FZD) varied between the two runs of each trial.

Evaluation metric. As previously defined, *location observation* is the portion of the engagement area visible from a location and *cumulative observation* is the sum of the location observation values for all the deployment locations selected for the entities of a unit. Cumulative observation was used to compare the performance of the two algorithms. Values for cumulative observation were computed by summing location observation values for the selected deployment locations for each of the entities.

The location observation values were obtained from the ModSAF terrain analysis tool, a feature of the ModSAF user interface that includes an area intervisibility function that graphically displays the area visible from a given point within a given circle, based on the intervisibility values for a grid of points within the circle. The ModSAF terrain tool was slightly modified so as to sum the intervisibility values within a given polygonal area. The sum so computed gives the location observation value for a given point relative to the given polygonal area. When that given polygonal area is an engagement area the intervisibility value sums returned by the terrain tool for a set of locations can be used as the location observation values. Those values were accumulated into a value for cumulative observation, relative to the engagement area, and compared between two deployments for the same engagement area.

Note that the modification did not change the underlying intervisibility process used by the ModSAF terrain tool; rather it simply summed the intervisibility values found to be within a polygon. The point is that the calculation of the cumulative observation metric is based on the ModSAF terrain tool, not on the observation value used by the FZD algorithm, making it an unbiased metric for comparing the algorithms.

4.3. Experiment results

Overall results. Table 1 summarizes the results of the 5 trials. The HOP and FZD cumulative observation values cannot be compared from trial to trial because they depend on the size of the engagement area, the number of features in the engagement area, and the ModSAF map scale when the terrain tool was used. However, they can be compared for the two runs of a trial, and show that the FZD algorithm outperformed the HOP algorithm on every trial.

Table 1. Cumulative observation values for the experiment trials

		Cumulative observation		
Trial #	**Trial name**	**HOP**	**FZD**	**Ratio, FZD/HOP**
(1)	Tree lines	8,950.75	21,855.58	2.44
(2)	Avenue of approach	10,929.68	13,536.80	1.24
(3)	Canopies	5,640.73	11,600.36	2.06
(4)	Road junction	7,856.39	16,118.99	2.05
(5)	River crossing	5,613.41	15,136.29	2.70

The ratio of the cumulative observation values FZD/HOP for each trial is also given. It can be compared from trial to trial. It shows that the difference in performance varied by trial, ranging from a low of 1.24 to a high of 2.70. The average ratio is approximately 2.10, so in general terms the FZD algorithm performed about twice as well as the HOP algorithm over all of the trials.

No perceptible difference in execution time between the algorithms was observed, so the execution times were not measured. Both algorithms ran in a few seconds.

Trials. Five trials were used to compare the two algorithms. Each was given a name descriptive of the terrain area used for the trial: (1) Tree lines, (2) Avenue of approach, (3) Canopies, (4) Road junction, and (5) River crossing. Trials (1) and (2), for which figures are given, are described in some detail; the descriptions for trials (3), (4), and (5) are abbreviated.

Figures 2 and 3 show the deployment locations chosen by the HOP and FZD algorithms for trial (1) Tree lines. The company was positioned within a loose cluster of tree lines on a company battle position that runs northeast to southwest and extends approximately 1200 meters. The engagement area is to the northeast of the battle position, is approximately 1500 ´ 1500 meters in extent, and is bordered on the right and divided across the center by more tree lines. The terrain was generally flat, though the company battle position was slightly elevated relative to the engagement area. This trial was intended to test the performance of the algorithms at avoiding the tree lines near the battle position and finding locations that saw between the tree lines in the engagement area. The HOP algorithm selected locations off the evenly spaced defaults for only 3 entities. The left subunit battle position generated by the FZD algorithm is relatively large because of generally poor observation over much of its length, but some cover locations were found within its battle area along the forward crest of the slight hill and the FZD algorithm chose those as deployment locations. Interestingly, in the small center subunit battle position the FZD algorithm used the default locations, due to the lack of cover and concealment and generally good observation of the engagement area in that battle area.

Figures 4 and 5 show the deployment locations chosen by the HOP and FZD algorithms for trial (2) Avenue of approach. A major road ran northeast to southwest between moderately dense clusters of tree lines and elevation rises, forming an obvious military avenue of approach. The company was positioned among tree lines on a hilltop at the southwest end of the avenue of approach, so as to block movement towards a major road junction just behind it. The company battle position ran northwest to southeast and was approximately 1000 meters long. The engagement area was to the northeast of the company battle position and is approximately 1000 x 2000 meters in extent. This trial was intended to determine if the algorithms would take advantage of the increased observation offered by certain locations on the hilltop near the battle position. The HOP algorithm generally used the default locations. The FZD algorithm fared slightly better, taking advantage of some covered locations on the hilltop behind the battle position with good observation of the engagement area.

In figures 3 and 5 it sometimes appears as if the tanks are touching or overlapping. That appearance is an illusion due to the fact that the tank icons are not shown at the same scale as the map. The algorithm includes a minimum distance between the selected deployment locations as a parameter; in these experiments it was 5 meters.

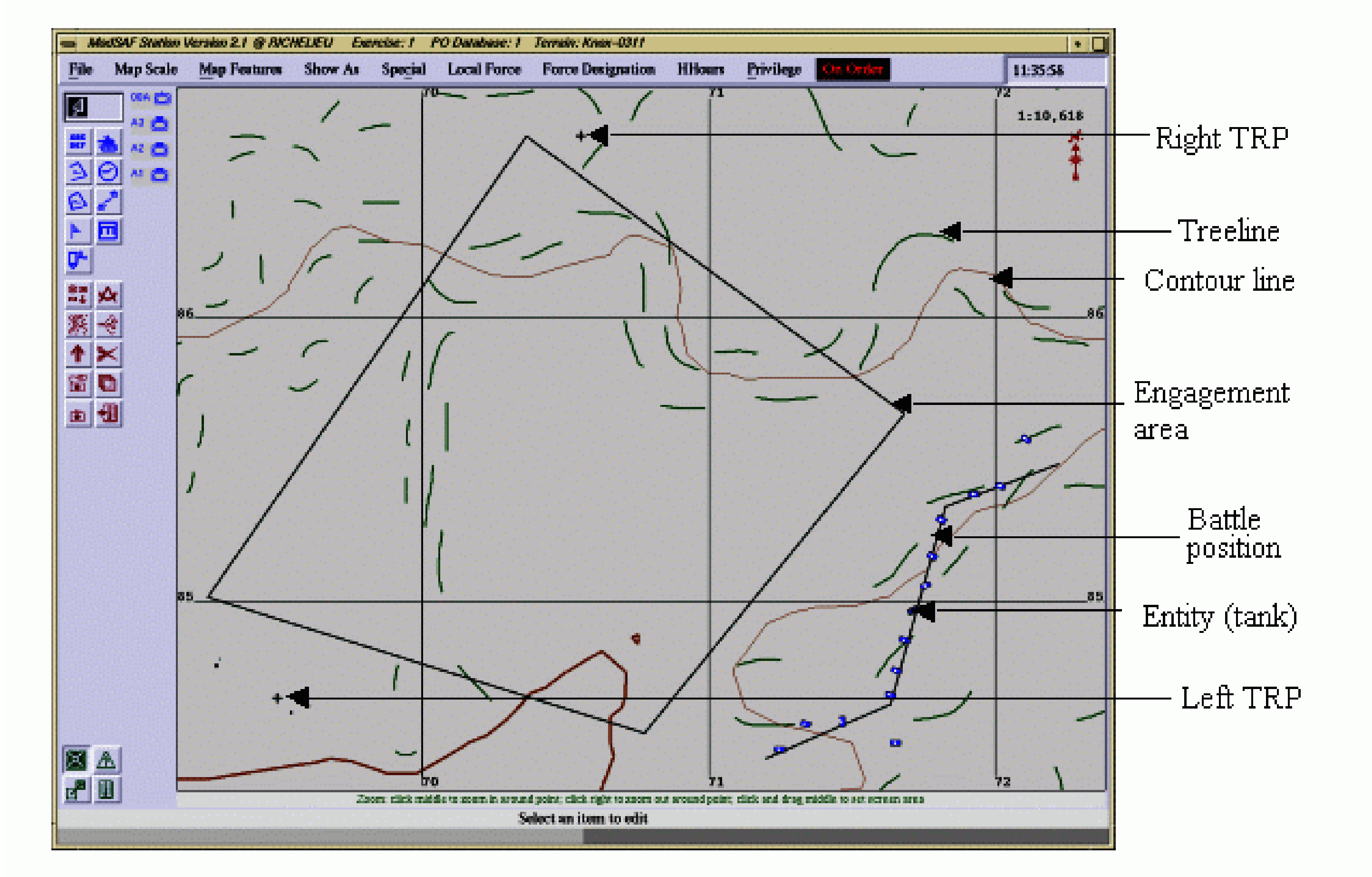


Figure 2. Hasty Occupy Position deployment for trial (1).

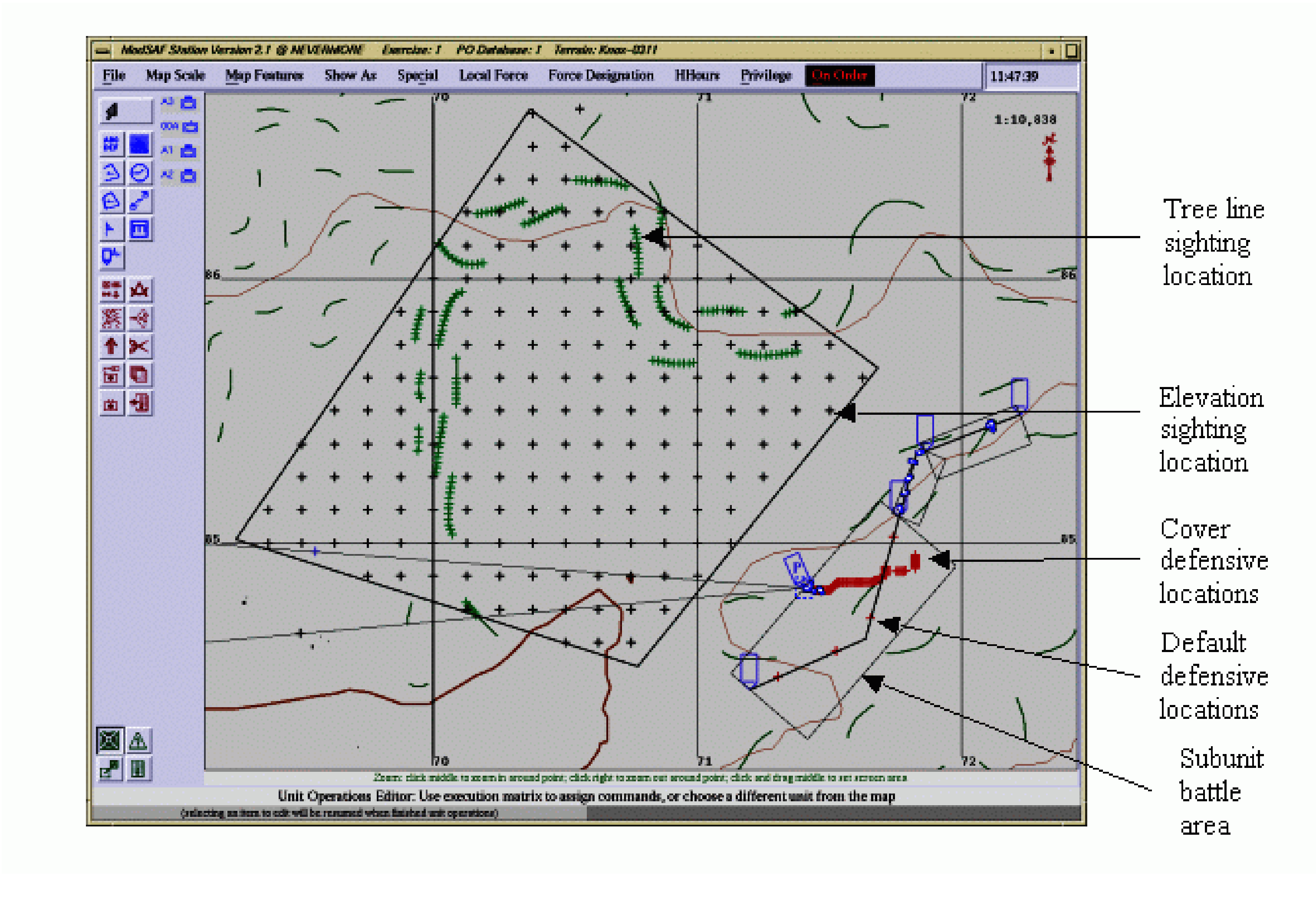


Figure 3. Fire Zone Defense deployment for trial (1).

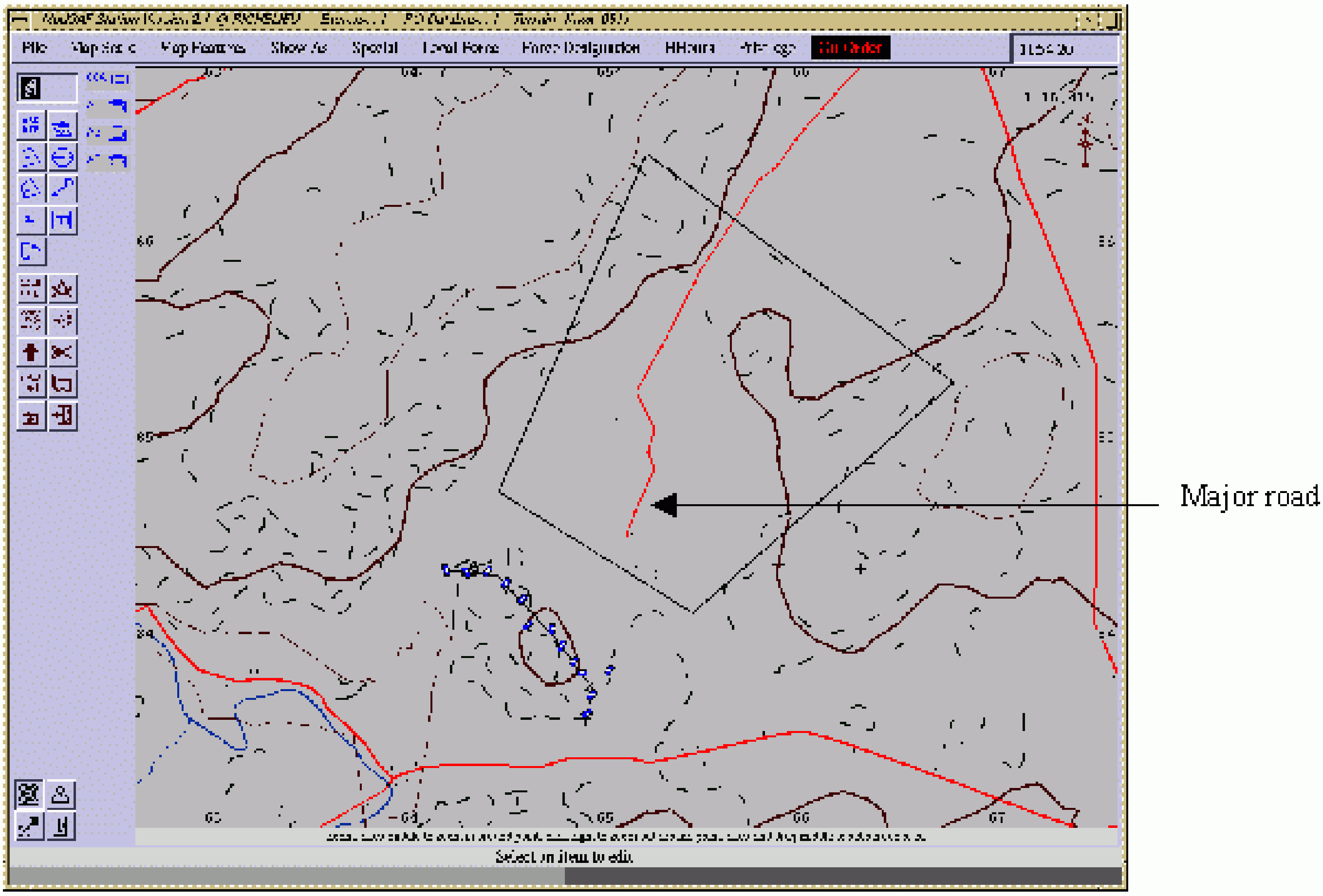


Figure 4. Hasty Occupy Position deployment for trial (2).

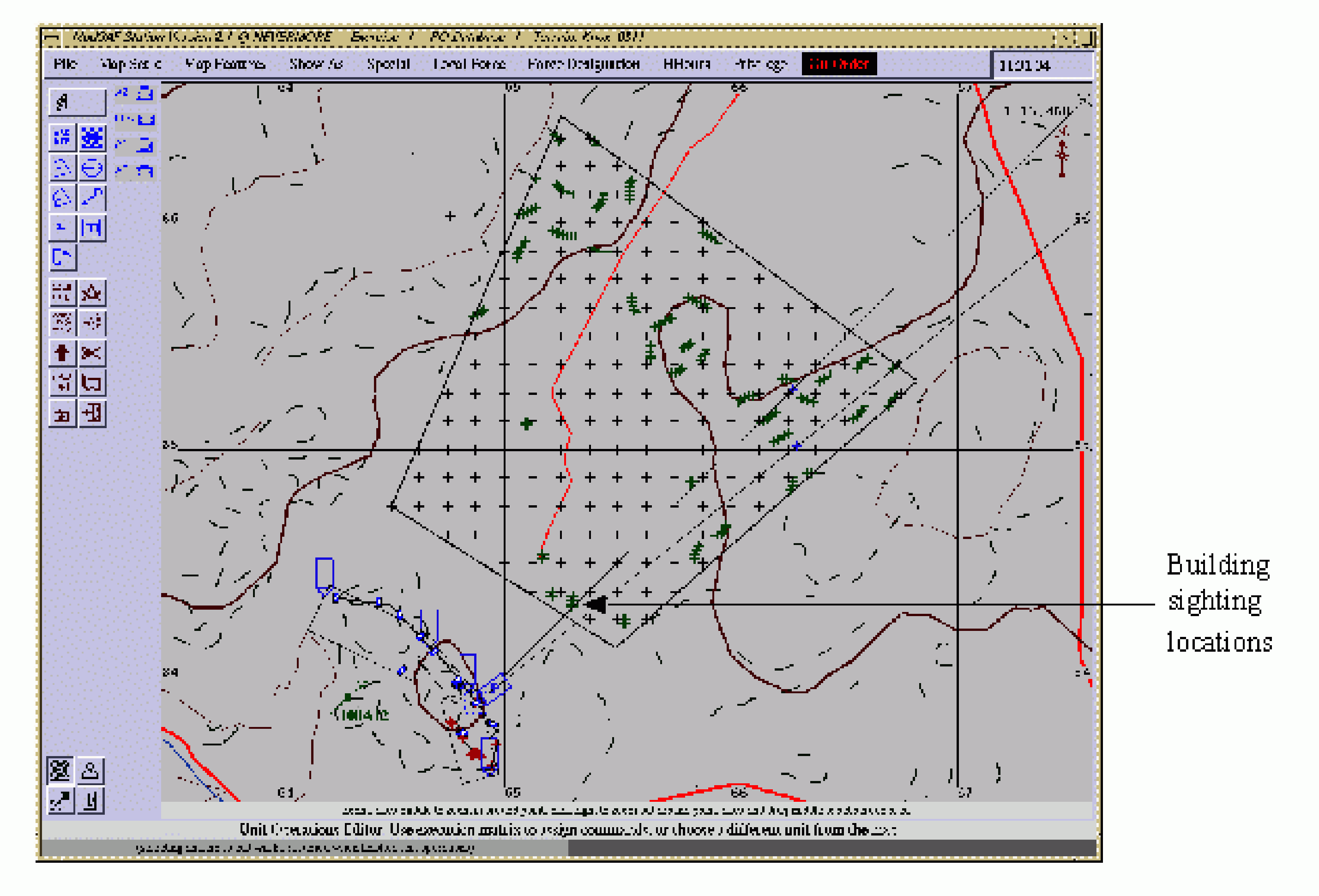


Figure 5. Fire Zone Defense deployment for trial (2).

Trial (3) Canopies had a relatively small engagement area located on flat, nearly featureless terrain between three large canopies. The FZD algorithm found better locations within the subunit battle areas behind the battle position. For trial (4) Road junction, the company was positioned adjacent to a major road along the northwest slope of a ridge that runs northeast-southwest, ending near the left end of the company battle position. The FZD algorithm deployed entities on covered locations on top of the ridge behind the battle position in the battle area of the right platoon. In trial (5) River crossing, the company is positioned on the southwest side of a river valley that ran northwest to southeast.

The HOP algorithm deployed entities where they could not observe the engagement area because of canopies and elevation slopes, whereas the FZD algorithm avoided the canopies and found good deployment locations along the forward crest in the left subunit battle area.

Overall comments. The FZD algorithm improves upon the HOP algorithm at two levels. At the company level, it partitions the unit battle position into subunit battle positions in a way that concentrates more entities in areas where better observation is available, while the HOP algorithm partitions the battle position based on length. This concentration is according to defensive doctrine: "The defender seeks to mass the effects of overwhelming combat power ..."[20] Second, at the platoon level, it avoids clearly suboptimal deployment of entities to locations that cannot observe the engagement area because of blocking terrain and instead finds good deployment locations from among the available defensive locations. Recall that both the HOP and the FZD algorithms are using the unmodified ModSAF cover and concealment finding algorithm to generate defensive locations. The difference is that FZD generates defensive locations by subunit, rather than by entity, and it chooses from among them based on observation, rather than by proximity to the default location.

Note that in two ways the FZD algorithm sacrifices optimum performance relative to the cumulative observation evaluation metric in order to gain military tactical effectiveness. First, entity sighting range is considered when computing observation values for subunit battle position partitioning and for selecting entity deployment locations, but it is not considered by the ModSAF terrain tool when computing the observation metric. In the latter case, the terrain analysis tool measures the visible portion of the engagement area from a location without regard to range. The FZD algorithm could have been implemented so as to likewise omit range from its observation value approximations and thereby achieve better values for the cumulative observation metric, but at the risk of deploying entities where they would get credit for observing portions of the engagement area they could not actually observe. We chose not to do so. Second, the reduction of the weights of observed sighting locations during the greedy optimization at the platoon level is intended to encourage broader, and more militarily effective, coverage of the engagement area by reducing the importance of sighting locations each time they are observed. Of course, that tends to reduce performance relative to the cumulative observation metric because the metric measures the portion of the engagement area observed without regard to how many entities observe the same area. Eliminating the weight reduction step from the FZD algorithm would have boosted its cumulative observation values. Again, we chose not to do so.

Conclusions

The existing HOP algorithm and our FZD algorithm are intentionally identical in functionality; they both defensively deploy the entities of a company sized unit, they both consider cover and concealment, and they both take as input a battle position and an engagement area. However, the experiment results show that with respect to the cumulative observation metric the FZD algorithm is approximately twice as effective as the HOP algorithm. This is a significant improvement and meets our goal in this task.

FZD outperforms HOP because it considers observation when generating a deployment. The use of geometry in choosing the sighting and defensive locations allows the FZD algorithm to consider observation when generating a deployment without excessive computation. The FZD algorithm as implemented also provides a framework for more sophisticated geometric analysis methods.

Acknowledgements

This project was partially supported by the UCF Institute for Simulation and Training (Petty and Franceschini) and the UCF Office of the Vice President for Research and Graduate Studies (Mukherjee). Old Dominion University supported the preparation of this paper for submission. That support is gratefully acknowledged. Lee Napravnik provided help with the ModSAF integration. Sumeet Rajput and Hai-Lin Nee also provided assistance.

References

1. A tutorial on CGF systems can be found in Mikel D. Petty, "Computer generated forces in Distributed Interactive Simulation," *Distributed Interactive Simulation Systems for Simulation and Training in the Aerospace Environment*, SPIE Critical Review 58 (Orlando FL, April 19-20 1995), 251-280.
2. For additional information on ModSAF the reader may refer to A.J. Courtemanche and A. Ceranowicz, "ModSAF Development Status," *Proceedings of the Fifth Conference on Computer Generated Forces and Behavioral Representation* (Orlando FL, May 9-11, 1995), 3-13.
3. For additional information regarding the CTDB format see J.E. Smith, "Recent Developments in ModSAF Terrain Representation", *Proceedings of the Fifth Conference on Computer Generated Forces and Behavioral Representation* (Orlando, FL: Institute for Simulation and Training, May 9-11, 1995), 375-381.
4. D.E. Mullally, *Personal communication* (1997).
5. U.S. Army, *Operations, U. S. Army Field Manual No. 100-5* (U.S. Government Printing Office, 1976).
6. Ibid.
7. U. S. Army, *Operations, U.S. Army Field Manual No. 100-5* (U.S. Government Printing Office, 1993).
8. Mullally, *Personal communication.*
9. E.C. O'Byrne, "Dismounted Infantry: Indispensible to the Virtual Battlefield," *Proceedings of the 15th Interservice/Industry Training Systems and Education Conference* (Orlando, FL, November 19-December 2, 1993), 783-791.
10. U. S. Marine Corps, *Ground Combat Operations, United States Marine Corps Operational Handbook 6-1*, (U.S. Government Printing Office, 1988).

11. D.M. Keirsey, J. Krozel, and D.W. Payton, "Scale-Space Representations for Flexible Automated Terrain Reasoning," *Proceedings of the U.S. Army Symposium on Artificial Intelligence Research for Exploitation of the Battlefield Environment* (El Paso TX, November 15-16, 1988), 108-118.
12. T.M. Cronin, "Allocating Sensor Envelope Patterns to a Map Partitioned by Territorial Contours," *Proceedings of the U.S. Army Symposium on Artificial Intelligence Research for Exploitation of the Battlefield Environment* (El Paso TX, November 15-16 1988), 65-78.
13. M.R. Hieb, G. Tecuci, J.M. Pullen, A. Ceranowicz, and D. Hille, "A Methodology and Tool for Constructing Adaptive Command Agents for Computer Generated Forces," *Proceedings of the Fifth Conference on Computer Generated Forces and Behavioral Representation* (Orlando FL, May 9-11 1995), 135-146; D. Hille, M.R. Hieb, G. Tecuci, and J.M. Pullen, "Abstracting Terrain Data Through Semantic Terrain Transformations," *Proceedings of the Fifth Conference on Computer Generated Forces and Behavioral Representation* (Orlando FL, May 9-11 1995), 355-365.
14. S. Rajput and H. Tu, "Terrain Analysis for Determining Tactical Positions," *Unpublished manuscript* (Institute for Simulation and Training, 1997).
15. Courtemanche and Ceranowicz, "ModSAF Development Status."
16. M.J. Longtin, "Cover and Concealment in ModSAF," *Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation* (Orlando FL, May 4-6 1994), 239-247.
17. D.R. Van Brackle, M.D. Petty, C.D. Gouge, and R.D. Hull, "Terrain Reasoning for Reconnaissance Planning in Polygonal Terrain," *Proceedings of the Third Conference on Computer Generated Forces and Behavioral Representation* (Orlando FL, March 17-19 1993), 285-305.
18. U. S. Army, *Operations, U. S. Army Field Manual No. 100-5* (U.S. Government Printing Office, 1986).
19. Mikel D. Petty, "Computational Geometry Techniques for Terrain Reasoning and Data Distribution Problems in Distributed Battlefield Simulation", *Ph.D. Dissertation*, School of Computer Science (University of Central Florida, 1997).
20. *U.S. Army Field Manual No. 100-5* (1993).

MIKEL D. PETTY is Chief Scientist of the Virginia Modeling, Analysis and Simulation Center at Old Dominion University. He has been performing and leading modeling and simulation research since 1990. He received a Ph.D. from the University of Central Florida in 1997, an M.S. from UCF in 1988, and a B.S. from the California State University Sacramento in 1980, all in Computer Science. His research interests are in distributed simulation interoperability, computer generated forces, and applied computational geometry; he has published over 70 papers in those areas. Address: Virginia Modeling, Analysis & Simulation Center, Old Dominion University, 7000 College Drive, Suffolk VA 23435. tel.: (757) 686-6210. E-mail: mpetty@vmasc.odu.edu.

ROBERT W. FRANCESCHINI is a Senior Research Computer Scientist at the Institute for Simulation and Training and a Visiting Assistant Professor in the School of Electrical Engineering and Computer Science at the University of Central Florida. He has performed research in distributed simulation, computer generated forces, multi-resolution simulation, data compression, and graph theory. He has over 30 published papers in those areas. Dr. Franceschini received a B.S. in Computer Science from the University of Central Florida in 1992 and a Ph.D. in Computer Science from UCF in 1999. Address: Institute for Simulation and Training, University of Central Florida, 3280 Progress Drive, Orlando FL 32826-0544. tel.: (407) 658-5519. E-mail: rfrances@ist.ucf.edu.

AMAR MUKHERJEE is a Professor of Computer Science in the School of Electrical Engineering and Computer Science at the University of Central Florida. Professor Mukherjee received the D.Phil. (Sc.) degree from the University of Calcutta in 1962 at the Institute of Radiophysics and Electronics. Since then his research has been focused on VLSI design, algorithms, and architectures. He has previously held faculty positions at the University of Iowa, Montana State University, and Princeton University. Professor Mukherjee is a Fellow of the IEEE and served two terms as Editor of the journal *IEEE Transactions on Computers*. Address: School of Electrical Engineering and Computer Science, University of Central Florida, 4000 Central Florida Boulevard, Orlando FL 32816-2362. tel.: (407) 823-2763. E-mail: amar@cs.ucf.edu.

BACK TO TOP

A Terrain Reasoning Algorithm for Defending a Fire Zone

Mikel D. Petty, Robert W. Franceschini and Amar Mukherjee

Abstract: Distributed simulation is an approach to building large-scale simulation models from a set of independent simulator nodes communicating via a network. The U.S. Army uses distributed simulation systems for both training and analysis. Those systems include both crewed simulators and computer generated forces (CGF) systems; the latter use software, rather than human crews, to generate the behavior of entities in the simulated battlefield. CGF systems must include algorithms for all of the tactical behaviors that are needed for the simulation. One such tactical behavior is "Fire Zone Defense". An algorithm for this behavior must select defensive deployment locations on the terrain for the individual entities (e.g., tanks) of a unit (e.g., a company) to effectively defend an assigned engagement area. The entities of the unit then move to those locations.

We developed a new algorithm for the behavior. It combines a geometric terrain analysis algorithm, which creates a weighted graph representation of the terrain, with a greedy optimization algorithm that operates on that graph. The algorithm was compared experimentally with a previously existing algorithm for the same behavior. The comparison used a metric that measured the cumulative observation of the engagement area from the selected locations of the defending entities. Under that metric the new algorithm consistently outperformed the existing algorithm, with an average ratio of performance over 2. The execution speeds for the two algorithms were approximately the same.

Author: **Valentine Penev**
Title: **Fuzzy Controller for Antitank Wire Guided Missile Simulator with Direct X SDK**
Year of issuance: **1999**
Issue: **Information & Security. Volume 3, 1999**
Hard copy: **ISSN 1311-1493**

FUZZY CONTROLLER FOR ANTITANK WIRE GUIDED MISSILE SIMULATOR WITH DIRECT X SDK

Valentine PENEV

Table Of Contents:

---

1. Introduction

The conditions of modern high technology engagement demand the best from operators. Some of the warriors ordered to active duty with the guided weapons had little or no launch training, making the use of simulation system crucial in gaining guidance experience. The missile launcher to operational deployment and combat is turned into simulation training system. The operator simultaneously can train under realistic war fighting conditions in the same unit. He actually will operate to maintain or enhance proficiency, especially in anti-armored engagements. Creation of the similar simulator is complex a task, where different skill have to be used.

Many studies have been performed on the regulator synthesis of an antitank wire guided missile, taking in mind possible countermeasures of mobile targets. Flexible features of a fuzzy regulator give a possibility to design pitch and yaw loops of missile with (line-of-sight) LOS guidance. In semi-automatic systems the operator only tracks the target, his task is to keep the optical center line of his sight pointing at the required point of missile impact.

In the first part of this paper the notations of the missile models are listed. In part 2 the missile model is yielded in the respect of three-dimensional pursuit problem. The effect of fin servo saturation is discussed in part three. The design of the fuzzy controller is described in the part 4, and effects of changes of membership functions and fuzzy inference rules related to the time-to-go, miss distance and other parameters are discussed in part 5. Part six is devoted to software issues. In the final section of the paper the main features of the simulator for antitank wire guided missile are discussed.

Notations

In this paper the following notations are used:

P, Q, R	Roll, pitch and yaw rate to the axes b_x, b_y and b_z
Φ	Body-axes roll angle

Θ	Body-axes pitch angle
Ψ	Body-axes yaw angle
$(X,Y,Z)^T$	Position vector of the missile center of mass with respect to the inertial frame
$(U,V,W)^T$	Velocity vector of the missile with respect to the body frame (m/s)
α, β	Attack angle and sideslip angle (rad)
δ_q, δ_r	Elevator deflation angle and rudder deflation angle
T	Thrust
M	Missile mass (kg)
S	missile wing reference area (m^2)
V_m	Magnitude of the missile velocity
M_x, M_y, M_z	External torques to the directions b_x, b_y and b_z
F_x, F_y, F_z	External forces to the directions b_x, b_y and b_z
C_{Fx}, C_{Fy}, C_{Fz}	Total aerodynamics force coefficients to the directions b_x, b_y and b_z
C_{Mx}, C_{My}, C_{Mz}	Total moment coefficients to the directions b_x, b_y and b_z
A_x, A_y, A_z	Acceleration along the directions b_x, b_y and b_z at center of mass
G_x, G_y, G_z	Forces of gravity to the directions b_x, b_y and b_z
I	Moment of the inertial tensor of the missile
M	Mach number
D	Reference length (m)
QS	Dynamic pressure
σ_z, σ_y	Misalignment to the central line
K_{sl}, k	Seeker stabilization gain
K_1,k_2,k_3	Guidance loop gains
T_1, T_N	Seeker and noise filter time constant, respectively

2. Missile Model

A conceptual medium range wire guided missile is considered where missile dynamics having six degrees of freedom are provided. Figure 1 shows the pitch guidance loop, which is typical for "twist and steer" and has rate and acceleration feedback type autopilot.[1,2,3] The yaw guidance loop is almost the same as the pitch channel[2,3] except that the pitch and the yaw change their places.

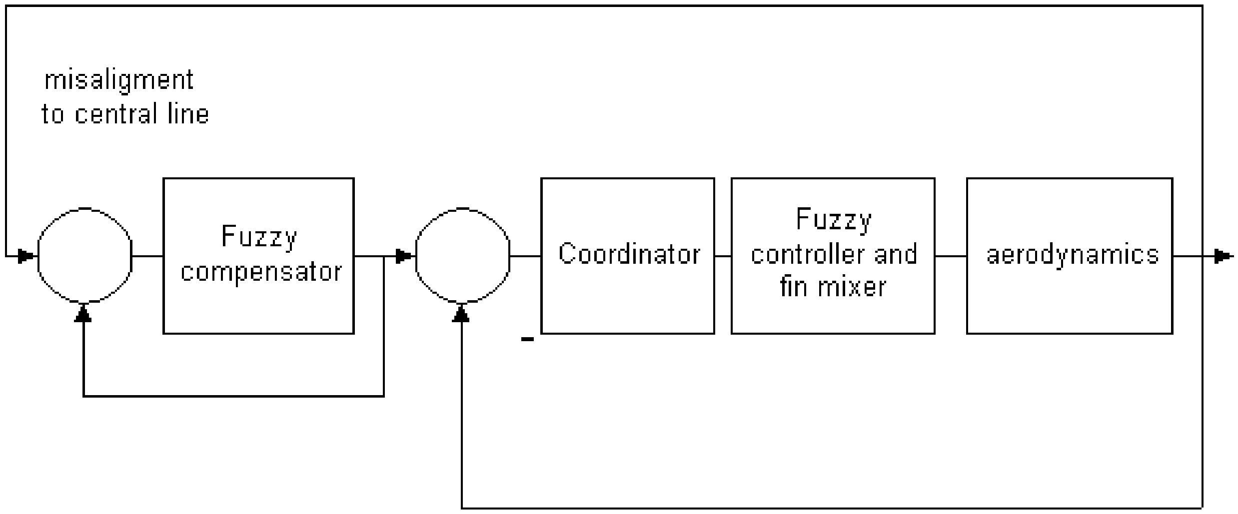


Figure 1: Missile pitch guidance loop

The 6-DOF missile dynamic equations[1,2,3,6] are written in the following state space form

$$\dot{P} = \frac{I_{yy} - I_{zz}}{I_{xx}} QR + \frac{M_x}{I_{xx}}$$

$$\dot{Q} = \frac{I_{zz} - I_{xx}}{I_{yy}} RP + \frac{M_y}{I_{yy}}$$

$$\dot{R} = \frac{I_{xx} - I_{yy}}{I_{zz}} PQ + \frac{M_z}{I_{zz}}$$

$$\dot{\Phi} = P + (Q \sin \Phi + R \cos \Phi) \tan \Theta$$

$$\dot{\Theta} = Q \cos \Phi - R \sin \Phi$$

$$\dot{\Psi} = \frac{Q \sin \Phi + R \cos \Phi}{\cos \Theta}$$

$$\dot{U} = RV - QW + \frac{1}{m} F_x$$

$$\dot{V} = -RU + PW + \frac{1}{m} F_y$$

$$\dot{W} = QU - PV + \frac{1}{m} F_z$$

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = R_1(\Phi)R_2(\Theta)R_3(\Psi)\begin{bmatrix} U \\ V \\ W \end{bmatrix}$$ - where R_1, R_2 and R_3 are the rotational matrices

The external forces and torques caused by the control surfaces, aerodynamics and gravity forces are defined as follows:

$$F_x = C_{Fx} \cdot QS \cdot S + T + G_x$$

$$F_y = C_{Fy} \cdot QS \cdot S + T + G_y$$

$$F_z = C_{Fz} \cdot QS \cdot S + T + G_z$$

$$M_x = C_{Mx} \cdot QS \cdot S_d$$

$$M_y = C_{My} \cdot QS \cdot S_d$$

$$M_z = C_{Mz} \cdot QS \cdot S_d$$

Part of normal values are listed as follows: m = 11.5 kg; Vm = 186.0 m/s; Tm = 580 N; ht0 = 0.50 m.

3. Fin Mixer

This section explores regions of instability and general problems of fin mixer.[2,3] Semi-automatic missiles often utilize a set of four fins to control pitch and yaw. Missiles of this type are typically guided by some form of LOS guidance (see figures 2 and 3). As a result the guidance loop gain is proportional to the range to the missile. The resulting fin commands sent from a classically developed autopilot often saturate the fin actuators, the rate, acceleration or position. When an actuator is driven into saturation, the control loop does not perform in the manner in which the control law has been designed. This can lead to instability or slow dynamical response. The internal square represents the area, where pitch and yaw commands may be implemented to decrease misalignment to center line.

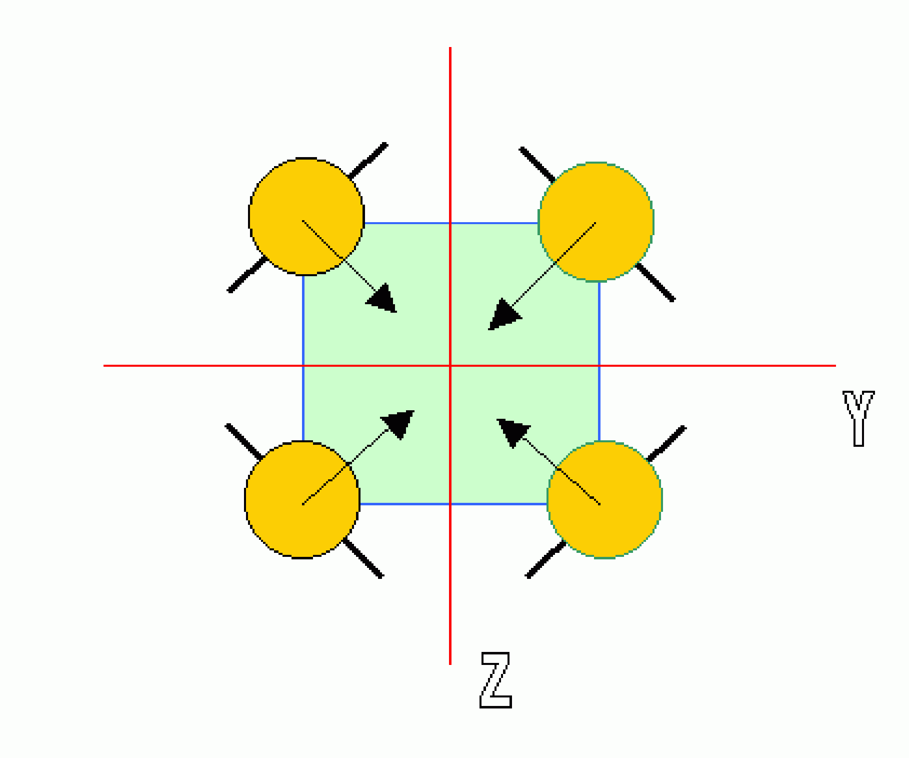


Figure 2: Rear view of representative twist and steer missile with 2 fins

This phenomenon is particularly troublesome for twist and steer missile which rely on the same fins for roll control and pitch control. During intercept the actuators are often driven into and/or acceleration limits due to large pitch commands caused by the LOS guidance and diminishing range. However, the fins must also be utilized to perform roll control to align the plane of maximum normal acceleration capability in the proper direction to pursue the sight. As a result of the saturation, roll control is lost momentarily and in some cases permanently, and the overall miss distance is increased.

This phenomenon is illustrated in figure 3, where the vector A represents the desired command in terms of combined pitch and yaw commands. The internal square represents the limits of capability of the fin actuators. For clarification purposes only position limitations are represented; however, rate and acceleration limits can be view in similar manner. Because the pitch command exceeds the actuator capability, the actuators are driven into saturation. As a result, not only is the pitch command not achieved, but the yaw command is also not accomplished as seen in vector B. For unstable missiles, this gain reduction can results in a temporary loss of stability. This instability is present as long as the fin actuators remain in saturation and often leads to large transient in roll error and increased miss distance.

It is desirable for a fin mixer to accommodate the limitations of the fins and perform the desired roll command represented by vector C. One way to minimize this effect is to reduce the guidance loop gain in the pitch axis. However, this results in performance loss for nominal intercepts.

The key to solving this problem is to develop a fin mixer which can anticipate actuator limitations and reduce the pitch gain prior to issuing fin commands, thus maintaining tight yaw control while providing maximum achievable pitch capability.

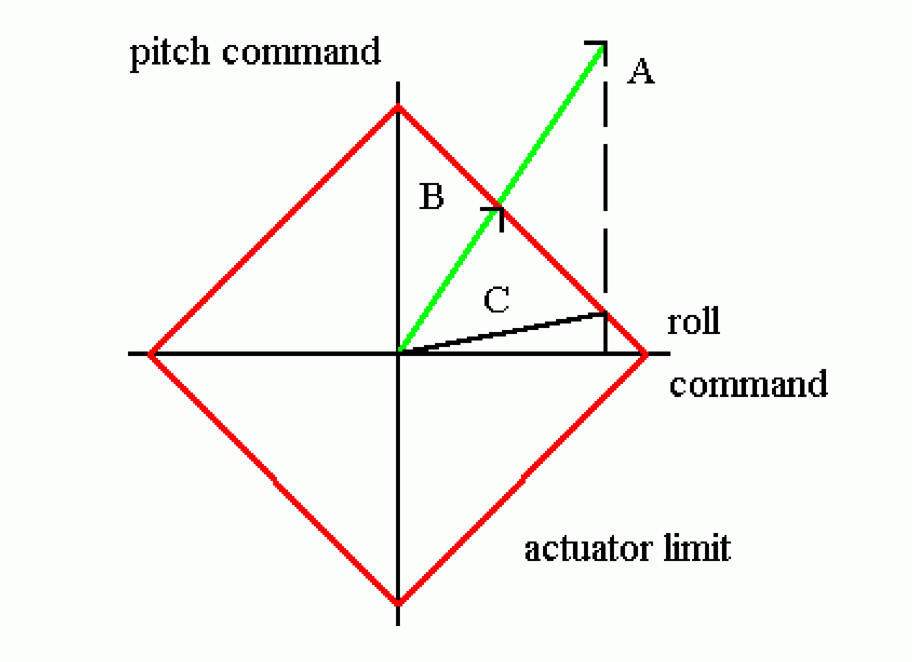


Figure 3: The effects of actuator limits

Let’s introduce the simple linear fin mixer common in most missile designs. Linear fin mixers translate the pitch and yaw commands from the control law to individual fin deflection commands in a linear fashion. This may be thought of as a simple matrix multiplication of the commands through a mixer matrix:

$$\begin{bmatrix} fin\ 1 \\ fin\ 2 \\ fin\ 3 \\ fin\ 4 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 & -1 \\ 1 & 1 \\ -1 & -1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} pitch\ \ command \\ yaw\ command \end{bmatrix}$$

$$u = 0.25\ MT\ \tau_d\ (1/4\ MM^T = 2I)$$

This technique is effective for a range of fin commands well within the actuator limitations, and it is with this mixer in mind that the classical control loops are designed. It is appropriate to use again fuzzy rules[4] to design a nonlinear fin mixers, which have identical mixing capabilities as the linear mixer within the limitation envelope of the fin actuators.

4. Fuzzy Control

LOS system can be called "three-point guidance" because there are target, missile and tracker or sight unit. The objective of the guidance loop is to move the missile as closely as possible on the line joining the tracker and the target. The concept is very simple, but there are some fundamental features, which limit the accuracy of LOS systems. The general characteristics which limit the short range semi-automatic systems are: missile "g" requirements in the respect of target motion, the limited beam width of tracker, uncertainty of the missile range.

The velocity control loop is suitable for systems designed to hit the stationary or slow moving targets, where small adjustments only are required by the operator. The basic idea is to incorporate the fuzzy IF-THEN rules into the control loop and to investigate how expert experience will guide highly nonlinear coupled object like missile.

The five input variables are used as follow: LOS pitch rate - *pr*; LOS yaw rate - *yr*; LOS pitch - *pp*; LOS yaw -*yy*; time-to-go - *t*. The LOS pitch rate and pitch, LOS yaw rate and yaw have six fuzzy allegiances - NB, NM, NS, PS, PM, PB. The membership functions have shape that is defined in the next equation and Table 1, where *u* may be *pr*, *yr*, *yy* and *pp*.

$$\mu(u)=\begin{cases} 0, & \alpha \le \alpha_1 \\ a_1\alpha + b_1, & \alpha_1 \le \alpha \le \alpha_2 \\ a_2\alpha + b_2, & \alpha_2 \le \alpha \le \alpha_3 \\ 0, & \alpha \ge \alpha_3 \end{cases}$$

Table 1. Maximal grades for MF (without magnifier).

	a1	**b1**	**a2**	**b2**	**α 1**	**α 2**	**α 3**
NB	-	-	-3/π	-2	-	-π	-2π /3
NM	3/π	3	-3/π	-1	-π	-2π /3	-π /3
NS	3/π	2	-3/2π	½	-2π /3	-π /3	π /3
PS	3/2π	½	-3/π	2	-π /3	π /3	2π /3
PM	3/π	-1	-3/π	3	π /3	2π /3	π
PB	3/π	-2	-	-	2π /3	π	-

The variable time-to-go has two fuzzy allegiances ZO and PS. The elevator deflation angle and rudder deflation angle are used as fuzzy output variables. For the commands δ_q and δ_r six fuzzy allegiances (NB, NM, NS, PS, PM, PB) are considered and the membership functions have the form of singleton.

The fuzzy engines for pitch and yaw loops[4,5] are shown on tables 2 through 5.

Table 2. Fuzzy rules for pitch loop - t is ZO.

pr/pp	NB	NM	NS	PS	PM	PB
PB	PM	PM	PS	NS	NM	NM
PM	PM	PM	PS	NS	NM	NM
PS	PB	PB	PS	NS	NB	NB
NS	PB	PB	PS	NS	NM	NM
NM	PM	PM	PS	NS	NM	NM
NB	PM	PM	PS	NS	NM	NM

Table 3. Fuzzy rules for yaw loop - t is ZO

yr/yy	NB	NM	NS	PS	PM	PB
PB	PM	PS	PS	NS	NS	NM
PM	PB	PM	PS	NS	NM	NB
PS	PB	PB	PM	NM	NB	NB
NS	PB	PB	PM	NM	NB	NB
NM	PB	PM	PS	NS	NM	NB
NB	PM	PM	PS	NM	NM	NM

Table 4. Fuzzy rules for pitch loop - t is PS.

pr/pp	NB	NM	NB	PB	PM	PB
PB	PB	PB	PB	NB	NB	NB
PM	PB	PM	PM	NM	NM	NB
PS	PM	PM	PM	NM	NM	NM

NS	PM	PM	PM	NM	NM	NM
NM	PB	PB	PM	NM	NB	NB
NB	PB	PB	PB	NB	NB	NB

Table 5. Fuzzy rules for yaw loop - t is PS.

yr/yy	NB	NM	NS	PS	PM	PB
PB	PB	PB	PB	NB	NB	NB
PM	PM	PM	PM	NM	NM	NM
PS	PM	PM	PM	NM	NM	NM
NS	PM	PM	PM	NM	NM	NM
NM	PM	PM	PM	NM	NM	NM
NB	PB	PB	PB	NB	NB	NB

5. Discussion of Controller

The result show that fuzzy controlled missile against high mobile target generally produces appropriate miss distance. But the increasing down distance leads to the very big miss distance. This investigation is a first step in using fuzzy control in the field of guided weapons and the investigation. The simulator consists of model of launcher unit, computer, first monitor mounted ahead of the sight, second monitor for controlling officer, keyboard and mouse.

6. Description of Direct X SDK

Direct3D® Immediate Mode application programming interface (API) is part of the three-dimensional (3-D) graphics component of DirectX®. Direct3D is designed to enable world-class game and interactive three-dimensional (3-D) graphics on a computer running Microsoft® Windows®. It provides device-dependent access to 3-D video-display hardware in a device-independent manner. Direct3D is a drawing interface for 3-D hardware.

Direct3D has two modes: Immediate Mode and Retained Mode. Retained Mode is a high-level 3-D API for programmers who require rapid development or who want the help of the Retained Mode built-in support for hierarchies and animation. Direct3D Immediate Mode is a low-level 3-D API that is ideal for developers who need to port games and other high-performance multimedia applications to the Windows operating system. Immediate Mode is a device-independent way for applications to communicate with accelerator hardware at a low level. Direct3D Retained Mode is built on top of Immediate Mode.

These are some of the advanced features of Direct3D:

- Switchable depth buffering (using z-buffers or w-buffers)
- Flat and Gouraud shading
- Multiple light sources and types
- Full material and texture support, including mipmapping
- Robust software emulation drivers
- Transformation and clipping
- Hardware independence
- Full support on Windows 95, Windows 98, and Windows 2000
- Support for the Intel MMX architecture

Developers who use Immediate Mode instead of Retained Mode are typically experienced in high-performance programming issues and may also be experienced in 3-D graphics. The best source of information about Immediate Mode is the sample code included with this SDK; it illustrates how to put Direct3D Immediate Mode to work in real-world applications. The world management of Immediate Mode is based on vertices, polygons, and commands that control them. It allows immediate access to the transformation, lighting, and rasterization 3-D graphics pipeline. If hardware is not present to accelerate rendering, Direct3D offers robust software emulation. Developers with existing 3-D applications and developers who need to achieve maximum performance by maintaining the thinnest possible layer between their application and the hardware should use Immediate Mode, instead of Retained Mode.

Direct3D Immediate Mode provides simple and straightforward methods to set up and render a 3-D scene. The key set of rendering methods are referred to as DrawPrimitive methods; they enable applications to render one or more objects in a scene with a single method call. For more information about these methods, see DrawPrimitive Methods. Immediate Mode allows a low-overhead connection to 3-D hardware. This low-overhead connection comes at a price; the designer must provide explicit calls for transformations and lighting, all the necessary matrices, and to determine what kind of hardware is present and what its capabilities are.

7. Building the Simulator

Wire guided antitank missile is a semi-automatic system. The operator only tracks the target, his task is to keep the optical center line of his sight pointing at the required point of missile impact. His sight is fitted with appropriate cross-wires and circles to assist him in aiming. Signals proportional to the angular misalignment of the missile from the center line of sight are processed by the ground unit (analog or digital computer) and are transmitted to the missile via wires. The processing of angular misalignment will include multiplication by term proportional to the missile range. The question is how accurately can an operator track the target.

To create the sensation of real combat conditions the computer simulation display mounted ahead of the sight is used. Because the system uses actual background terrain pictures, a variety of realistic environments from the desert to winter scene can be used. There are plenty of ground features such as buildings, rivers, threes and so on. The visual display simulation of weather, with particularly good representation of scud, fog is better than in much more expensive simulators. A personal computer controls the simulation system, automatically scores engagements and collects training management data. Audio equipment realistically replicates engine and weapon. Inputs to the computer come from instructor or operator.

The simulator for antitank wire guided simulator /ATGMS/ consists of launcher, computer, software package, sight control unit, computer, keyboard, mouse, second monitor (figure 4). This version is for indoor usage. Software package consists of simulator, scenario maker and viewer. ATGMS was designed into Direct3D Immediate Mode, which gives possibility to render the scene with 40000 vertices in real-time.

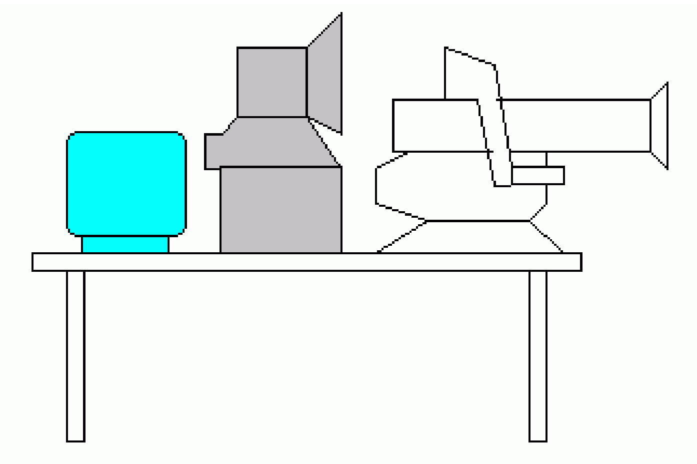


Figure 4: General view of ATGMS

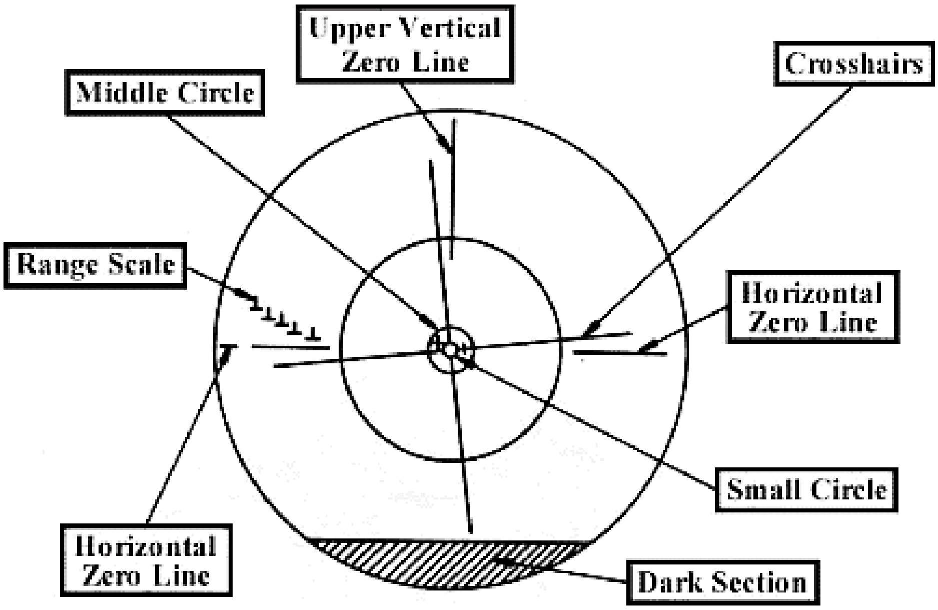


Figure 5: Sighting reticle

In the simulator of wire guided antitank missile the following objects can be defined:

- Terrain
- Stationary objects – buildings, trees, bushes, bridges
- Slow moving objects – tanks, armored vehicles, helicopters
- Fast moving objects – the missile
- Atmospheric effects – fog

The Environment3D window displays the view from the launch unit sight (figure 5). There are some objects in this view: scene with background terrain, moving and stationary targets, other graphical objects that are part of the current configuration. In the Environment3D window, we have to see the missile response to the commands from the vertical and horizontal controls. It is very easy to connect software application to the vertical and horizontal controls from the launching unit using the DirectX Input.

References

1. Babister A. W., *Aircraft stability and control* (1961).
2. Bodner V.A., *Control systems for aircrafts* (Moscow: Mashinostroenie, 1973).
3. Garnell, P., *Guided weapon control systems*, 2nd ed. (Oxford, England: Pergamon Press, 1980).
4. Driankov D., H. Hellendoorn, and M. Reinfrank, *An introduction to fuzzy control* (SpringerVerlag, 1997).
5. Valentine Penev, Petya Koprinkova, Georgi Georgiev, and Peter Petrov, "Fuzzy guidance controller for air to air missile," Presented during *HEMUS-96* (Plovdiv, Bulgaria, 1996).
6. John M. Rolfe, *Flight simulation*, (Cambridge: Cambridge University Press, 1988).

VALENTINE PENEV: Born 1960. M.Sc. (1986, Automatics and Telemechanics) from the Technical University of Tula, Russia. Ph.D. (1992, Fuzzy sets in digital control systems) from the Robotics and Control Systems Research Center at the Bulgarian Academy of Sciences. Currently Dr. Penev is Deputy Director for R&D of the Institute of Control and Systems Research at the Bulgarian Academy of Sciences. Senior Research Fellow with over forty publications in the field of intelligent technology engineering - fuzzy control, fuzzy reasoning, signal processing, and neuro-like

structures. Software realization of a flight simulator and the simulator "Fagot" for ATGMs under Windows95, Windows 3.1x and DOS for PC.
Address for correspondence: Institute of Control and Systems Research, Bulgarian Academy of Sciences, Acad.G.Bonchev Str., Bl.2, Sofia 1113, Bulgaria. E-mail: valpenev@iusi.bas.bg

BACK TO TOP

Information & Security. An International Journal
e-mail: infosec@mbox.digsys.bg

Fuzzy Controller for Antitank Wire Guided Missile Simulator with Direct X SDK

Valentine Penev

Keywords: missile, modeling, aerodynamic, fin mixer, fuzzy control, 3D simulation.

Abstract: This paper examines the main steps in the process of creating an antitank wire guided missile simulator. This principal task includes the following items: understanding the behavior of the simulated object, appropriate mathematical modeling and 3-dimensial visualization. The general features of a fuzzy controlled semi-automatic missile are studied and missile response against the mobile target is evaluated and discussed. The main characteristics of simulator for antitank wire guided missile designed with DirectX7 SDK are presented.

Authors: **Plamena Andreeva, George Georgiev;**
Title: **Fuzzy Control Based on Cluster Analysis and Dynamic Programming;**
Year of issuance: **1999**
Issue: **Information & Security. Volume 3, 1999**
Hard copy: **ISSN 1311-1493**

FUZZY CONTROL BASED ON CLUSTER ANALYSIS AND DYNAMIC PROGRAMMING

Plamena ANDREEVA and George GEORGIEV

Table Of Contents:

---

1. Introduction

Fuzzy control as an approach to nonlinear and complex control design has attracted a great deal of research interest in the past decade. The basic idea of the approach is to incorporate fuzzy IF-THEN rules into the control design, that is, fuzzy control combines two resources: input-output data and the experts' experience expressed by rules. Therefore, fuzzy control is always applied to the system, which is too complex to get the mathematical model precisely.

Fuzzy control systems are dynamic systems that estimate input – output functions without mathematical model and adaptively infer and generate fuzzy rules. We apply fuzzy control in two general ways. First one is to use only fuzzy controller. The second one is to use fuzzy rules like tuning machine for original controllers.

By complex non-linear objects the control of a process is nothing more than a linguistic model of the human operator strategy and is, as such, a decision model. Individual decision making in fuzzy environment (FE) is:

$$\mu_{D^\wedge}(x_{max}) = max_x \, min \{\mu_{G^\wedge}(x), \mu_{C^\wedge}(x)\},$$

where $x_{max\,i}$ is the maximizing decision. Here $\mu_{G^\wedge}(x)$ is the fuzzy goal, $\mu_{C^\wedge}(x)$ is the fuzzy constrain. We consider the discrete decision space (situations in which there are countable many actions from which the "optimal" has to be chosen). Utilities (cost function) are modeled as fuzzy sets, in the framework of linguistic variables.

1.1. Problem statement

The problem of incorporating expert knowledge is wide studied and founds applications in variety of control, signal processing and expert systems.[6, 10, 11] It can be considered as non-linear regression to approximate the unknown non-linear function based on the input–output data. If the input–output data are measurements of the inputs and outputs of an expert who is demonstrating on how to do a skilled job, then the collection of input–output data is a presentation of the expert's domain knowledge. The advantage of the fuzzy system approach compared with other non-linear regression methods is that fuzzy systems not only are universal approximators,[11] but also have clear physical interpretation for their structures and parameters so that the results can be interpreted in terms of fuzzy IF-THEN rules.

The basic problem in fuzzy control application is the analysis of the transmitted uncertainty from the premises to the conclusion. Cluster analysis of the input domain helps to a reasonable extent in the management of uncertainty in such situations.

This paper concerns fuzzy control for autonomous mobile system in 3D space. The associated information system is designed to be a tool for automated analysis and synthesis of closed loops for moving platforms and to store the results from possible training made by an expert and distributed via network. In order to determine the number of rules during the training process a clustering method is proposed. The number of rules is specified by the number of clusters of the input–output data and the parameters are determined from the a priory context dependant knowledge representation.

1.2. Current research

Fuzzy models are described by fuzzy rule of type (1) plus some additional parameters. Fuzzy models are just particular instances of the kind of non-linear non-parametric model with the advantage of providing the fuzzy rules as a way to describe some possible available prior knowledge.

$$\textit{If } x_1 \textit{ is } A_1 \textit{ and } x_2 \textit{ is } A_2 \textit{ and } \ldots \textit{ and } x_n \textit{ is } A_n, \textit{ then } y \textit{ is } B, \qquad (1)$$

where $A_1, A_2, \ldots, A_n$ and B_n are fuzzy sets. The input and output domain of such fuzzy system (shown on Fig.1) are determined in relation to the input and output universe of discourse *U* of fuzzy sets. A fuzzy set *A* whose support A_{sup} is the universe of discourse *U* with $\mu(x)=1$ is referred to as a fuzzy universe.

The objective of this paper is to perform theoretical analysis of the closed loops for moving objects by the mean of generating clustering method for extracting rules from the Distributed Information System (DIS). The tool for performing this analysis is applied to improve the performance of moving object control and off-line analysis of its efficiency. This research is related to the search of the suitable rule sets of the fuzzy classification system. Clustering method with triangular membership functions is presented and the case with Gaussian membership functions is considered.

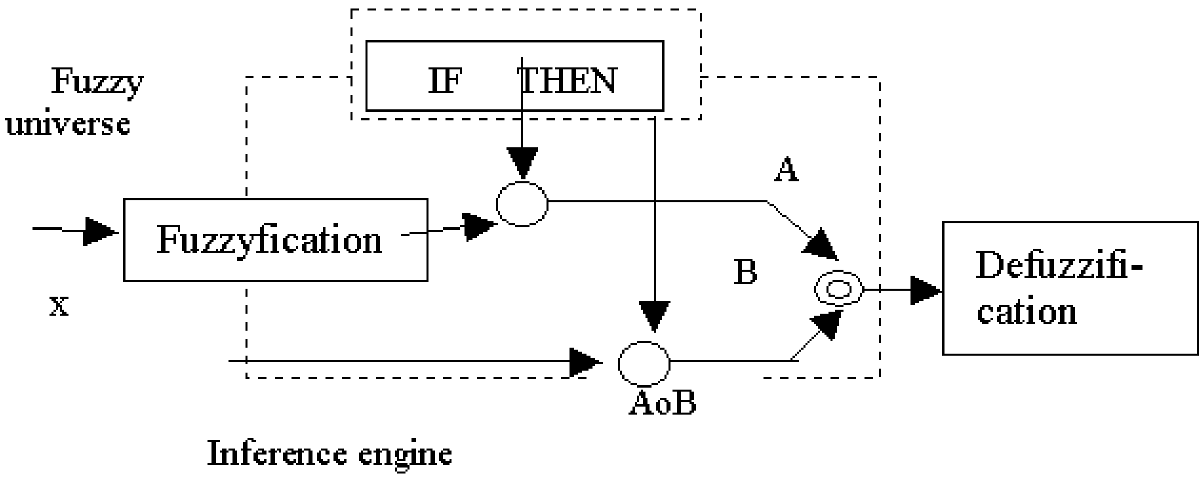


Figure 1. Basic structure of fuzzy control system.

The classification problem is the capability of the system to generate the required decision regions. This process helps in reducing the searching space and time while performing the optimal solution. Because representation of the rule base (R) by an analytical function is unrealistic and infeasible in general, it is customary to assume that the fuzzy sets involved in R have discrete and finite universes. For reasons of processing efficiency, it is easier to scale the process variables and control signals in fuzzy control algorithm. Real-time modification of a fuzzy rule on the basis of some self-organization scheme is possible.[4] Our fuzzy approach combines the power of fitting complex data from DIS and the possibility of structuring the knowledge by linguistic rules. More important, the analysis eventually leads to a new representation of the input data.

2. Analysis of the cluster method

Classification of a priori knowledge and identification of new data are central in the fuzzy control process. Fuzzy control for moving objects is characterised by model uncertainty and inequality model constraints. The proposed method in (Andreeva, et. al., 2000) of cluster analysis for such a system is used to classify the input data and to receive the rules.

Let us consider the problem as given the input –output data pairs:

$$(x^p; y^p),\ p=1,2,...N \qquad (2)$$

where $x^p \in U_x = [\alpha_1, \beta_1] \times ... \times [\alpha_n, \beta_n] \subset R^n$, $y^p \in U_y = [\alpha_y, \beta_y] \subset R$ and the data are generated by unknown nonlinear function or system $y = f(x)$. Here α_i and β_i are respectively the lower and upper bound of input – output domain, and U_i is universe of discourse for variable x_i. The goal is to design a fuzzy system $f(x)$ based on these N input – output pairs which approximate the unknown function or system $f(x)$.

If we assume to have two input – output pairs $(x^1; y^1)$ and $(x^2; y^2)$ first we have to define fuzzy sets to cover the input and output spaces. For $i = 1, 2$ there will be two fuzzy sets A_1^j and A_2^j , where $j = 1,2,...,N$ according to (2).The fuzzy sets have the following membership function:

$$\mu_{A_i^1}(x_i) = \mu_{A_i^1}(x_i; e_i^1, e_i^1, e_i^2),$$

$$\mu_{A_i^{j_i}}(x_i) = \mu_{A_i^{j_i}}(x_i; e_i^{j_i-1}, e_i^{j_i}, e_i^{j_i+1}), \qquad (3)$$

$$\mu_{A_i^{N_i}}(x_i) = \mu_{A_i^{N_i}}(x_i; e_i^{N_i-1}, e_i^{N_i}, e_i^{N_i+1}),$$

for $j_i = 2,3,..,N_i-1$ and x_i –input data; e_i^{ji} – average value; left boundaries j^{i-1} and right boundaries j^{i+1}. Here the triangular membership function $\mu\ A_i^j$ is defined as :

$$\mu_A(x; a, b, c_i) = \begin{cases} \dfrac{x-a}{b-a}, & a \le x \le b \\ \dfrac{c-x}{c-b}, & b < x \le c \\ 0, & \text{otherwise} \end{cases} \qquad (4)$$

If we assume to have two input – output pairs $(x^1; y^1)$ and $(x^2; y^2)$ first we have to define fuzzy sets to cover the input and output spaces.

$$\left| x^k - x_c^{l_k} \right| \le \left| x^k - x_c^l \right|, \qquad l = 1,2,..., M \qquad (5)$$

For $i = 1, 2$ there will be two fuzzy sets A_1^j and A_2^j , where $j = 1,2,...,N$ according to (2).The fuzzy sets have the membership function in (3).

Starting with the first input – output pair $(x^1; y^1)$ we establish a cluster center x_c^1 at x^1 and set $y_c^1(1)= y^1$, where the number of clusters $M=1$ and y_c^M equals to the centroid of input data. No other cluster exists at this time. For the *k-th* input –output pair (x^k, y^k), $k=2,3,..K$, there are M clusters with centres at $x_c^1, x_c^2, .., x_c^M$. In the next step find the nearest cluster x_c^l to x^k.

$$f_k(x) = \frac{\sum_{l=1}^{M} y_c^l(k) \prod_{j=1}^{2} \mu(x_j; x_{c,j}^l - r, x_{c,j}^l, x_{c,j}^l + r)}{\sum_{l=1}^{M} \prod_{j=1}^{2} \mu(x_j; x_{c,j}^l - r, x_{c,j}^l, x_{c,j}^l + r)} \qquad (6)$$

If the absolute value is greater than a selected radius r of first cluster, then establish x^k as a new cluster with center $x_c^{M+1} = x^k$ and set $y_c^{M+1}(k)=y^k$ and keep $y_c^l(k)=y_c^l(k-1)$. If the absolute value in (5) is less then the radius r, than the input point belongs to its closest cluster $k-1$ and compute the cluster's center $y_c^l(k)=y_c^l(k-1)$. The algorithm stops after the reach of last input-output data. We then construct the fuzzy system as:

$$\left| f(x) - f_k(x) \right| \leq kr \sum_{i=1}^{k} \sup(\min_{l} \left| x - x_c^l \right|) \qquad (7)$$

In the area outside the cluster the constant value of the nearest cluster center is used as the estimate. This is reasonable because nothing is known about the unknown function in the area not covered by the given data.

This method has no recursion and its simplicity allows soft clustering. It differs from C-Means clustering method [3] where the number of clusters is predefined. Thus, in contrast to prior research, the applied fuzzy clustering method becomes more suitable for off-line training analysis. The problem is sometimes called unsupervised learning (or self-organization). By unsupervised learning, the performance error on inferring procedure has no effect on the operation of the clustering algorithm. In unsupervised mode, the primary measure that affects the goodness of a cluster is the performance error.

2.1. Clustering method with triangular membership functions

We consider the clustering method with assumption that $f(x)$ is continuously differentiable in the whole unit of discourse U. If $f_k(x)$ is the obtained fuzzy system under the above assumption it holds that :

$$\mu_{A_i^j}(x_i) = \exp\left(-\frac{| x_i - e_i^j |^2}{\sigma^2}\right) \qquad (8)$$

in case that $x \in M$, some cluster and k is the number of input-output pairs. The difference between the considered data and the cluster center it belongs to is the Euclidean distance $d_x = min_l |x-x_c^l|$.

2.2. Method with Gaussian membership functions

The complete clustering method was shown in the previous section. Here the method with Gaussian membership function is presented. It is similar to that in section 2.1 with the exception that the membership functions are Gaussian. They are of the form (8).

Then the fuzzy system is constructed as:

$$f_k(x) = \frac{\sum_{l=1}^{M} y_c^l(k) \exp(-\frac{|x - x_c^l|^2}{\sigma^2})}{\sum_{l=1}^{M} \exp(-\frac{|x - x_c^l|^2}{\sigma^2})} \quad (9)$$

Instead of averaged value y_c^l, the linear interpolation of the data inside the cluster can be used to construct fuzzy system. By doing this, more accurate result may be obtained, but the computation is more complex.

3. Simulation

The control loops of moving objects were designed on the base of fuzzy control theory. Fuzzy control as an approach to nonlinear and complex control design has attracted a great deal of research interest in the past decade. The autopilot design for flying objects, which are highly nonlinear-coupled system, provides for potentially useful implementation of fuzzy control.

The proposed fuzzy algorithm in section *2.1* is applied to the autonomous mobile system. As input data we consider the experimental data collected during the training process. For modeling dynamic characteristics of moving objects we use at least six nonlinear equations which describe linear and angular positions of rigid body in 3D space with 6 degrees of freedom. Vehicle parameters (geometric and cinematic) are space vector (x_n, y_n, z_n), velocity ν , control angle θ and its azimuth α . The output variables are the control angle φ and vehicle velocity ν_n.

The altitude, airspeed, vertical speed and angular position are measured. The control of the aeroplane is done by IF-THEN rules. Fuzzy control combines two resources: input-output data and the experts' experience expressed by rules. The a priori information is given in the initial state of input data. The universe of discourse of angle φ is chosen [0, 360] and velocity ν_n is [0, 980]. For the partitioning of input data we assume that the fuzzy sets of φ and ν_n have symmetric triangular membership functions.

The fuzzy clustering method is performed as follow:

1. For each input variable X_i and output φ and ν_n we define fuzzy sets to cover the input and output spaces.

   For the 3 input variable we define N_1, N_2 and N_3 fuzzy sets A_1^j, A_2^j, A_3^j (j=1,2,..N_i) on each [α_i, β_i] with the triangular membership functions as in (3).

2. From one input-output pair we generate then one rule.

   For each input variable determine the fuzzy set A_1^j, A_2^j, A_3^j in which ν^p, θ^p and α^p (p=1,2,...N_i) has the largest membership value. Similarly define ν_n^p and φ^p such that $\mu \nu_n^p(y^p) >= \mu \nu_n(y^p)$ and $\mu \varphi^p(y^p) >= \mu \varphi(y^p)$ for p=1,2,...N_y. Finally the fuzzy IF- THEN rule is:

   If ν is A_1^j, and θ is A_2^j, and α is A_3^j,

   THEN ν_n is B_1^p and φ is B_2^p (10)

3. Degrees are assigned to each rule generated in 2).

   If there exist conflicting rules (with the same IF parts but different THEN parts) then assign a degree to each rule generated in 2.

   $$Deg\ (rule) = \Pi_i\, \mu_{Ai}(\nu^p, \theta^p \text{ and } \alpha^p) \,.\, \mu_{Bj}(\nu_n^p, \varphi^p) \quad (11)$$

where i =1,2,3, and j=1,2. Then keep only one rule from a conflict group that has the maximum degree.

4. The fuzzy rule base (R) will be created as :

Rule $^{(f)}$:

$$\begin{aligned} &\mathit{If}\ x_1\ \mathit{is}\ A_1{}^{j1},\ \mathit{and}\ x_2\ \mathit{is}\ A_2{}^{j2},\ \mathit{and}\ x_3\ \mathit{is}\ A_3{}^{j3,} \\ &\mathit{THEN}\ y\ \mathit{is}\ B^{(f)} \end{aligned} \tag{12}$$

with *f* index set of the rule base.

The condition $\Sigma\,\mu_{jc}=1$, where *i* is i-th input data and *c* is number of the cluster *i* belongs to, is not satisfied. This follows from the interpretation of membership values in the *c-th* partitioning. The boundaries between the classes are really fuzzy – more data belong to one cluster with different μ .

4. Algorithm and decision-making method

An algorithm for fuzzy control of an autonomous mobile system in 3D space is described. The algorithm is synthesized on the basis of the Bellman and Zadeh decision-making method in vague environment. The introduction of a goal and a constraint are presented as fuzzy sets. The autonomous vehicle attains the goal by avoiding the constraint by dynamic programming in fuzzy environment. The coordinates of the model movement are determined by maximum-minimum composition. A maximizing decision is defined as an intersection in the space of alternatives of membership functions of a fuzzy goal and a fuzzy constraint at which the membership function of a fuzzy decision attains its maximum.

4.1. Initial Conditions

The problem involves finding the best possible time (optimal) registration path of the autonomous mobile system from the initial state to the goal by avoiding the constraint. The goal and the constraint do not change theirs coordinates during the process of decision making. The goal is presented by as a physical previous given point at which the autonomous mobile system has to land or cross. The constraint is presented as an object, which the autonomous mobile system has to avoid. It is given the following initial conditions:

$\mathbf{x_0, y_0}$	- coordinates of the initial state
$\mathbf{x_g, y_g}$	- final coordinates
$\mathbf{x_c, y_c}$	- coordinates of the constraint
$\mathbf{v_{ok}}$	- average initial velocity value of the model by air
$\gamma_{\mathbf{ok}}$	- average initial value of the model flight-path angle
$\Psi_{\mathbf{ok}}$	- initial model azimuth angle by north
$\mathbf{h_{ok}}$	- initial altitude of the goal
$\Delta\Psi$	- increment of the model azimuth angle
$\Delta\Phi$	- increment of the model bank angle

Δt	- increment of the real time of movement
Φ_{max}	- maximum value of the model bank angle
w_x	- wind acceleration along the x-axis
w_y	- wind acceleration along the y-axis
L	- lift
D	- drag
m	- mass
g	- acceleration of gravity
ρ	- air density
s	- model reference area
α	- angle of attack
α_0	- zero lift angle
C_D, C_{D0}	- drag coefficient and zero-lift drag coefficient
C_L, $C_{L\alpha}$	- lift coefficient and its partial derivative of α
wins	- width of the matching window
k_x, k_y	- coefficient reflecting the slope of the selected membership function

The mass of the autonomous mobile system is concentrated in one point. The flexible structure of the model is described by the function $\boldsymbol{f}^L(\boldsymbol{v}, \gamma, \Phi)$ and $\boldsymbol{f}^D(\boldsymbol{v}, \gamma, \Phi)$. The model of the autonomous mobile system is described by the following nonlinear derivative equations:

$$\dot{v}_t = \frac{-D}{m_t} - g \cdot \sin \gamma_t - w_x \cdot \cos \gamma_t - w_k \cdot \sin \gamma_t \qquad (13)$$

$$\dot{\gamma}_t = \frac{L}{m_t v_t} \cos \Phi - \frac{g}{v_t} \cos \gamma_t - \frac{\left(w_x \cdot \sin \gamma_t - w_y \cdot \cos \gamma_t \right)}{v_t} \qquad (14)$$

$$\dot{\Psi}_t = \frac{L}{m_t v_t \cos \gamma_t} \cdot \sin \Phi \qquad (15)$$

$$\dot{x}_t = v_t \cos \gamma_t \cos \Psi_t \qquad (16)$$

$$\dot{y}_t = v_t \cos \gamma_t \sin \Psi_t \qquad (17)$$

$$\dot{h}_t = v_t \sin \gamma_t \tag{18}$$

Where: $L = \frac{1}{2} \rho \cdot v_t^2 \cdot s_t \cdot C_L \cdot f^L(v, \gamma, \Phi)$ (19)

$$C_L = C_{L\alpha}(\alpha - \alpha_0) \tag{20}$$

$$D = \frac{1}{2} \rho \cdot v_t^2 \cdot s_t \cdot C_D \cdot f^D(v, \gamma, \Phi) \tag{21}$$

The system of derivative equations is integrated by Oilier. Thus the following system of recurrent equations is obtained:

$$v_{k+1} = v_k - \frac{D}{m}.\Delta t - (g.\sin \gamma_k + w_x.\cos \gamma_k + w_y.\sin \gamma_k).\Delta t \tag{22}$$

$$\gamma_{k+1} = \gamma_k + \frac{L.\cos\Phi}{m.v_k}.\Delta t - \frac{g.\cos \gamma_k - w_x.\sin \gamma_k + w_y.\cos \gamma_k}{v_k}.\Delta t \tag{23}$$

$$\Psi_{k+1} = \Psi_k + \frac{L}{m.v_k.\cos \gamma_k}.\sin \Phi.\Delta t \tag{24}$$

$$x_{k+1} = x_k + v_k.\cos \gamma_k.\cos \Psi_k \cdot \Delta t \tag{25}$$

$$y_{k+1} = y_k + v_k.\cos \gamma_k.\sin \Psi_k \cdot \Delta t \tag{26}$$

$$h_{k+1} = h_k + v_k.\sin \gamma_k \cdot \Delta t \tag{27}$$

The states of the system are x_k, y_k, v_k, h_k, Ψ_k, γ_k. The system can be controlled by the control angle Φ_k. and the velocity v_k. The dynamic features of the object may be given by limitations of the control angle Φ_k depending on its velocity v_k. The limitations are described for a specific case as follows:

$$\Phi = \begin{cases} \Phi_{max}, \Phi \geq \Phi_{max} \\ \Phi, -\Phi_{max} < \Phi < \Phi_{max} \\ -\Phi_{max}, \Phi \leq -\Phi_{max} \end{cases}$$

where: $\Phi_{max} = -55^\circ.\frac{v}{10}; \quad +55^\circ.\frac{v}{10}$

The membership function of the fuzzy goal might be presented as an attractive function, which has a maximum at the final point. Its kind is described as follows:

$$\mu_g(x, y) = e^{-k_x(x_g - x)^2 - k_y(y_g - y)^2} \tag{28}$$

The membership function of the fuzzy constraint might be presented as a repulsive function and it is described by the following form:

$$\mu_c(x, y) = 1 - e^{-k_x(x_c - x)^2 - k_y(y_c - y)^2} \tag{29}$$

In the algorithm the following variables and areas are used:

i, j, k, min,max	- work variables
$\mathbf{x_k(wins)}$	- work area of possible x-coordinates of the movement trajectory
$\mathbf{y_k(wins)}$	- work area of possible y-coordinates of the movement trajectory
$\mathbf{\Psi_k(wins)}$	- work area of possible azimuth angles of the movement trajectory
$\mathbf{\mu_g(wins)}$	- work area of the membership function values of the model states by the goal
$\mathbf{\mu_c(wins)}$	- work area of the membership values of the model states by the constraint
$\mathbf{i_{max}}$	- variable for the width of the matching window at which the system states is obtained maximizing intersection
$\mathbf{\nu_k, \nu_{ok}}$	- variables for the model velocity
$\mathbf{\gamma_k, \gamma_{ok}}$	- variables for the model flight-path angle
$\mathbf{h_k, h_{ok}}$	- variable for the model altitude
$\mathbf{x_{ok}}$	- variable for the model x-coordinate
$\mathbf{y_{ok}}$	- variable for the model y-coordinate
$\mathbf{\Psi_{ok}}$	- variable for the model azimuth angle
K	- number of the one-stage decision process
$\mathbf{x_{tr}(K)}$	- work area of optimal x-coordinates for the whole movement trajectory
$\mathbf{y_{tr}(K)}$	- work area of optimal y-coordinates for the whole movement trajectory
$\mathbf{\Psi_{tr}(K)}$	- work area of optimal azimuth angles for the whole movement trajectory
$\mathbf{\Phi_c}$	- variable for the control value of the bank angle which has been computed for the optimal azimuth angle
$\mathbf{\Psi_t}$	- variable for the current computed azimuth angle
$\mathbf{d\Psi_t}$	- variable for the current module difference of the azimuth angle
$\mathbf{d\Psi_{min}}$	- variable for the minimum module difference of the azimuth angle

Subscripts:

K	- new value of the model states of the decision process
o, ok	- initial or old value of the model states of the decision process
t	- current value

4.2. Control Algorithm

Step 1 It is given the following initial conditions: x_0, y_0 - coordinates of the initial state x_g, y_g - final coordinates x_c, y_c - coordinates of the constraint v_{ok}, γ_{ok}, Ψ_{ok}, h_{ok}, $\Delta\Phi$, Δt, Φ max , w_x, w_y, L, D, m, g, ρ , s, ,α , α_0, C_D, C_{D0}, C_L, $C_{L\alpha}$, wins, k_x, k_y $\Delta\Psi = \frac{2.\pi}{wins}, x_{ok} = x_g, y_{ok} = y_g, k = 1$	**Step 2** i = 1 $\Psi_k(i) = \Psi_{ok}$

Step 3
 	
$x_k(i) = x_{ok} - v_{ok}.\cos\gamma_{ok}.\cos(\Psi_{ok}).\Delta t$
$y_k(i) = y_{ok} - v_{ok}.\cos\gamma_{ok}.\sin(\Psi_{ok}).\Delta t$
 	
$\mu_g(i) = e^{-k_x(x_0 - x_k(i))^2 - k_y(y_0 - y_k(i))^2}$
 	
$\mu_c(i) = 1 - e^{-k_x(x_c - x_k(i))^2 - k_y(y_c - y_k(i))^2}$
	Step 4
	
	IF (i ≠ wins) THEN
	
	i = i + 1
	
	$\Psi_k(i) = \Psi_k(i-1) + \Delta\Psi$
	
	IF $(\Psi_k(i) > 2.\pi)$ THEN
	
	$\Psi_k(i) = \Psi_k(i) - 2.\pi$
	
	$\Psi_{ok} = \Psi_k(i)$
	
	go to Step 3

Step 5
 	
I = 1
 	
Max = 0
	Step 6
	
	IF $(\mu_g(i) \le \mu_c(i))$ THEN
	
	min = $\mu_g(i)$
	
	j = I
	
	ELSE
	
	min = $\mu_c(i)$
	
	j = I

Step 7 IF (min ≥ max) THEN max = min Imax = j	**Step 8** IF (i ≠ wins) THEN i = i + 1 go to Step 6
Step 9 $x_{tr}(k) = x_k(i_{max}),\quad y_{tr}(k) = y_k(i_{max}),$ $\Psi_{tr}(k) = \Psi_k(i_{max}),\quad x_{ok} = x_k(i_{max}),$ $y_{ok} = y_k(i_{max}),\quad \Psi_{ok} = \Psi_k(i_{max})$	**Step 10** IF (($x_{ok} \neq x_0$) or ($y_{ok} \neq y_0$)) THEN k = k + 1 go to Step 2
Step 11 $\Psi_t = \frac{L}{m.v_{ok}.\cos\gamma_{ok}}.\sin(-\Phi_{max}).\Delta t$ $d\Psi_{min} = \lvert\Psi_t - \Psi_{tr}(k)\rvert$ $\Phi_c = -\Phi_{max}$, $\Phi = -\Phi_{max} + \Delta\Phi$	**Step 12** $\Psi_t = \frac{L}{m.v_{ok}.\cos\gamma_{ok}}.\sin(\Phi).\Delta t$ $d\Psi_t = \lvert\Psi_t - \Psi_{tr}(k)\rvert$
Step 13 IF $(d\Psi_t \leq d\Psi_{min})$ THEN $d\Psi_{min} = d\Psi_t$ $\Phi_c = \Phi$	**Step 14** IF $(\Phi \neq \Phi_{max})$ THEN $\Phi = \Phi + \Delta\Phi$ go to Step 12

Step 15
 	
$\gamma_k = \gamma_{ok} + \frac{L.\cos\Phi_c}{m.v_{ok}}.\Delta t -$
$\frac{g.\cos\gamma_{ok} + w_x.\sin\gamma_{ok} - w_y.\cos\gamma_{ok}}{v_{ok}}.\Delta t$
 	
$v_k = v_{ok} - \frac{D}{m}.\Delta t - (g.\sin\gamma_{ok} +$
$w_x.\cos\gamma_{ok} + w_y.\sin\gamma_{ok}).\Delta t$
 	
$h_k = h_{ok} + v_{ok}.\sin\gamma_{ok}.\Delta t$
 	
$\gamma_{ok} = \gamma_k,\ v_{ok} = v_k,\ h_{ok} = h_k$
 	
The model movement is directed by the
	Step 16
	
	IF (k ≠ 0) THEN
	
	k = k - 1
	
	go to Step 11
	
	ELSE
	
	End

calculated bank angle Φ_c, x_{ok}, y_{ok}, v_{ok}, h_{ok}, Ψ_{ok}, and γ_{ok}.	

4.3. Description of the Algorithm

The decision uses the method of dynamic programming in fuzzy environment. The width value of the matching window is a variable named wins. The fuzzy environment consists of a goal and a constraint. The matched pairs of membership function values of the goal and the constraint are compared in the range of the matching window. Finding the best time (optimal) registration path of the model is connected with the decision of the system of nonlinear derivative equations. The purpose of the presented algorithm is to demonstrate a fuzzy method for determination of the trajectory of the dynamic object, which is described by nonlinear derivative equations. The problem of finding the optimal registration path of the model is a multistage decision process. The original multistage (K - stage) decision process is replaced as K one-stage processes. It is the principle of optimality on which dynamic programming is based. That is why the model equations (22) - (27) are presented in a recurrent form. The model state a for k+1 one-stage decision process may be expressed in the following way:

$$X_{k+1} = X_k + Max(\mu_g(1) \wedge \mu_c(1), \mu_g(2) \wedge \mu_c(2), \ldots, \mu_g(wins) \wedge \mu_c(wins)) \quad (30)$$

where:

X_k - general symbol for the model state $(\mu_g(i), \mu_c(i))$ - matched pair of membership function values of the goal and the constraint$1 \leq i \leq wins$ - range of alteration

In the beginning the algorithm uses the reverse problem of dynamic programming. The model is moved from the goal to the initial state. The optimal movement trajectory is calculated in an Off-line mode. The next initial condition is admitted: The model velocity v_{ok} and the model flight-path angle γ_{ok} accept an average initial value. Equations (25) and (26) are used. The model azimuth angle is altered from 0 to 2π for each one-stage decision process. The increment of the azimuth angle $\Delta\Psi$ is calculated by the following expression:

$$\Delta\Psi = \frac{2.\pi}{wins} \quad (31)$$

The pair coordinates $x_k(i)$ and $y_k(i)$ are calculated. It begins with the given initial azimuth angle Ψ_{ok}:

$$\begin{aligned} x_k(i) &= x_{ok} - v_{ok}.\cos\gamma_{ok}.\cos(\Psi_{ok}).\Delta t \\ y_k(i) &= y_{ok} - v_{ok}.\cos\gamma_{ok}.\sin(\Psi_{ok}).\Delta t \end{aligned} \quad (32)$$

The calculated pair coordinates are replaced in the membership functions of the fuzzy goal and the fuzzy constraint. The possible coordinates of the next point from the movement trajectory and the corresponding azimuth angles are stored in the three area $x_k(wins)$, $y_k(wins)$ and $\Psi_k(wins)$. By using a maximum-minimum composition such model coordinates are determined at which the intersection of matched pairs of membership functions of the goal and the constraint obtains maximum value. Those values of the azimuth angles are stored at which the coordinates get maximum value. This is being done for each k one-stage decision process. The policy function is determined on the basis of the azimuth angle by maximum-minimum composition and it is expressed as follows:

$$\Psi_{ok} = \pi(Max(\mu_g(i_{max}) \wedge \mu_c(i_{max})) = \pi(Max(\mu_{g\cap c}(i_{max}))) = \Psi_k(i_{max}) \quad (33)$$

where:

$\mu_g(i_{max}) = e^{-k_x(x_0 - x_1(i_{max}))^2 - k_y(y_0 - y_1(i_{max}))^2}$ - is the membership function value by the goal at which the system state has obtained a maximizing intersection

$\mu_c(i_{max}) = 1 - e^{-k_x(x_c - x_1(i_{max}))^2 - k_y(y_c - y_1(i_{max}))^2}$ - is the membership function value by the constraint at which the system state has obtained a maximizing intersection

imax – is the width value of the matching window at which the system state has obtained a maximizing intersection.

The optimal coordinates of the movement trajectory and the corresponding angle of the azimuth are stored in three areas $x_{tr}(k)$, $y_{tr}(k)$ and $\Psi_{tr}(k)$. Their maximum dimension is equal to the number of the one-stage decision processes K. The algorithm uses the direct problem of dynamic programming in an On-line mode after it has attained the initial state. The bank angle is altered from Φ_{max} to Φ_{max} in equation (24) for each one-stage process. The average initial values of the model velocity ν_{ok} and the flight-path angle γ_{ok} are substituted for the first calculation of the azimuth angle Ψ_t. The difference by module $d\Psi$ min is formed between the calculated azimuth angle Ψ_t and the calculated azimuth angle from the reverse problem of dynamic programming $\Psi_{tr(k)}$. The minimum module difference $d\Psi_{min}$ is looked for each alteration of the increment of the bank angle $\Delta\Phi$. This value of the bank angle Φ_c is stored at which the module difference $d\Psi_{min}$ has obtained minimum value. The calculated bank angle Φ_c is replaced in equation (23) and the new flight-path angle is calculated. The new model velocity and altitude are calculated in equations (22) and (27). The system states are found. The problem of finding the best possible time (optimal) registration path of autonomous mobile system is decided. The model movement is directed by the calculated bank angle Φ_c.

5. Conclusion

This paper focuses on fuzzy control of a class of nonlinear systems, which are characterized by model uncertainty and inequality model constraints. It deals with an autonomous moving system for control of flying objects and a distributed information system, designed as a tool for automated analysis and synthesis of closed loop. Different clustering methods are explored and compared. The associated Intelligent Information System (IIS) is designed to store the results from possible training made by an expert and distributed via network. The paper considers cluster analysis for such a system, based on Bezdek’s fuzzy cluster method (FCM). The proposed method is used to classify the input data and to receive the rules.

The number of rules is specified by the number of clusters. It is used to classify the input data and to receive the rules for DIS. This rules can then be used for the design of rule-based intelligent systems. When a new unknown input is added the fuzzy classifier scheme proceeds and executes the rule to find a new fact. This allows stepwise refinement in DIS. We have discussed a possible analysis toward this objective. At the end some simulation results are given. The results showed that the key factors for accuracy are the resolution of the input space covering and the a priori information.

This has to be further explored. In addition, it would be interesting to develop statistical methods for diagnostics and learning from fuzzy – neural nets. Then the individual fuzzy rules can be learned using an error back-propagation algorithm.

An example of fuzzy control for autonomous mobile system in 3D space is explored and the results from the decision using the method of dynamic programming in fuzzy environment are shown. The synthesized algorithm guides an autonomous vehicle in 3D space which pursues an object and evades an obstacle. The fuzzy control is based on determination of a maximizing decision by using dynamic programming. The maximizing decision is defined as a point in the space of alternatives at which the membership function of a fuzzy decision attains its maximum value. The purpose of the presented algorithm is to demonstrate a fuzzy method for determination of the trajectory of the dynamic object.

A system of functional equations can be decided by using dynamic programming and appropriate membership functions for fuzzy environment. The optimal registration path is computed for an autonomous mobile system in 3D space. The fuzzy control has been presented for model movement along the optimal path. In this sense the above-described approach can be used for some other functional applications as well. The fuzzy environment could be expanded, too.

Acknowledgement

The work on this paper was supported by the Bulgarian Academy of Sciences, project "Development of Moving Platform simulator".

References

1. Andreeva, P., N. Tchingov and R. Michel, "Cluster Analysis for Fuzzy Control Systems," *Proc. of IFAC Conference CSD'2000* (Bratislava, Slovak Republic, 12-15 June 2000).
2. Bellman, R.E. and L.A. Zadeh, "Decision making in fuzzy environment," *Management science* 17, 4 (1970), 141-164.
3. Bezdek J.C. and Pal S.K, *Fuzzy Models for Pattern Recognition* (NJ: IEEE Press, 1992).
4. Driankov D. and H. Hellendoorn, *An Introduction to fuzzy control* (Berlin: Springer Verlag, 1997).
5. Georgiev, G., "Algorithm for fuzzy control of autonomous mobile robot," *INCON'97* (Sofia, Bulgaria, October 13-15, 1997), 53-56.
6. Hirota, K., ed., *Industrial Applications of Fuzzy Technologies* (Tokyo: Springer Verlag, 1994).
7. Kubota, T. and H. Hashimoto, "A strategy for collision avoidance among moving obstacle for a mobile robot," *IFAC Control Conference* (1992), Vol. 5, pp. 103-108.
8. Penev, V., "Heurestic Decision-Making Training System," *EUFIT'94* (Aachen, Germany, September 20-23, 1994), Vol. 2, pp.660-664.
9. Sakoe H. and S. Chiba, "A dynamic programming approach to continuous speech recognition," *Proc. Int. Congres on Acoustics* (Budapest, Hungary, 1971), Paper 20, C-13.
10. Wang, L.-X., *Adaptive fuzzy Systems and Control* (Englewood Cliffs, N.J.: Prentice Hall, 1994).
11. Wang, L.-X. and L.M. Mendel, "Generating Fuzzy Rules by Learning from Examples," *IEEE Trans. on Systems, Man and Cybernetics* 22 (1992), 1414-1427

PLAMENA TSANOVA ANDREEVA holds a M.Sc. degree in electronics from the Technical University of Sofia, Bulgaria (1986), and graduated the Informatics course in Theoretical Computer Science at the Institute of Informatics from the University of Fribourg, CH (1998). She is interesting in Knowledge Based Systems and Artificial Intelligence. IEEE , EWISH member. Mrs. Andreeva is currently a research associate at the Department of Knowledge Based Control Systems, Institute of Control and System Research of the Bulgarian Academy of Sciences. E-mail: plamena@iusi.bas.bg.

GEORGE LUBOMIROV GEORGIEV: M.Sc.(1985, Computer Technologies) from the Technical University of Sofia, Bulgaria. Research and design works in the field of intelligent technologies engineering (signal processing, fuzzy control, fuzzy reasoning, neuro-like structures); Projects of robot control systems; sensors & systems based on modern control theory; simulators. Mr. Georgiev is currently a research associate at the Department of Knowledge Based Control Systems, Institute of Control and System Research of the Bulgarian Academy of Sciences. E-mail: joro@iusi.bas.bg.

BACK TO TOP

Information & Security. An International Journal
e-mail: infosec@mbox.digsys.bg

Fuzzy Control Based on Cluster Analysis and Dynamic Programming

Plamena Andreeva and George Georgiev

Keywords: Fuzzy Control, Knowledge Based Systems, Clustering algorithm, Decision-making, Learning in Fuzzy Environment, Dynamic programming.

Abstract: This paper focuses on fuzzy control of a class of nonlinear systems, which are characterized by model uncertainty and inequality model constraints. The associated Intelligent Information System (IIS) is designed to store the results from possible training made by an expert and distributed via network. The paper considers cluster analysis for such a system, based on Bezdek's fuzzy cluster method (FCM). The proposed method is used to classify the input data and to extract the rules.

An example of fuzzy control for autonomous mobile system in 3D space is explored and the results from the decision using the method of dynamic programming in fuzzy environment are shown. The synthesized algorithm guides an autonomous vehicle in 3D space which pursues an object and evades an obstacle. The fuzzy control is based on determination of a maximizing decision by using dynamic programming. The maximizing decision is defined as a point in the space of alternatives at which the membership function of a fuzzy decision attains its maximum value. The purpose of the presented algorithm is to demonstrate a fuzzy method for determination of the trajectory of the dynamic object.

Authors: **Petya Koprinkova, Valentine Penev**
Title: **Dynamical Behavior of Fuzzy Logic Based Velocity Control Autopilot with Respect to Changes in Linguistic Variables Membership Functions Shape**
Year of issuance: **1999**
Issue: **Information & Security. Volume 3, 1999**
Hard copy: **ISSN 1311-1493**

DYNAMICAL BEHAVIOR OF FUZZY LOGIC BASED VELOCITY CONTROL AUTOPILOT WITH RESPECT TO CHANGES IN LINGUISTIC VARIABLES MEMBERSHIP FUNCTIONS SHAPE

Petya KOPRINKOVA and Valentine PENEV

Table Of Contents:

---

1. Introduction

Recently, fuzzy control has become one of the most attractive areas of fuzzy set theory applications.[3,10] As it provides an effective way to approximate inexact nature of human thinking, this theory is an appropriate tool for converting linguistically expressed expert's knowledge about a given process control into mathematically defined control strategy. This is made by means of the so called linguistic values, which are fuzzy sets over given universe of discourse of corresponding linguistic variable. The membership functions, which describe this linguistic values, as a rule, are defined by some bell-shaped function. Several nonlinear definitions of this functions are known,[2] but most researchers prefer to use simple linear triangular or trapezoidal functions. Usually in fuzzy controller synthesis membership functions shape is defined in advance and then only their position in the universe of discourse is tuned.

It is shown[1,8] that the bell shaped membership functions appear to be more appropriate for some cases of fuzzy control systems because the shape of the membership functions influences the shape of the fuzzy controller actions surface[6,7] and, hence, the dynamics of the system under control,[8] as well as its stability.[9]

In the present paper the investigation of the influence of nonlinearities in membership functions shape on fuzzy logic based velocity control autopilot dynamics is investigated.

2. Problem Statement

The concept of "velocity control"[4] is associated with manual control systems where the operator tracks both the missile and the target and attempts to bring target and missile into alignment by issuing commands to the missile, usually by means of a specially designed thumb controller and a command link to the missile. If no autopilot is used then a constant demand - either up-down or left-right - will result in the missile developing a steady lateral acceleration.

Consider the action necessary by the operator to make a correction in position in the yaw plane. To put in a left correction a good operator will put over his controller to the left until half the "error" is eliminated and will then reverse his controller. Ideally, transients in the missile should be so short that the operator is not really aware of their presence. A great deal of anticipation and training is required to develop such a skill; the operator is using his own intelligence to supply the necessary amount of phase advance.

Consider a missile carrying a position gyro which feeds back a voltage proportional to yaw angle. The situation now is that the net signal into the rudder is the difference between the command and the achieved angular position of the missile. When the missile is pointing in the desired direction the rudder will automatically be returned to the central position and the missile will continue to fly straight in this direction. Such a system of controlling the trajectory of a missile is called "velocity control." The opinion is widely held that the training of operators is much easier with velocity control than with acceleration control. Another advantage of a gyro in the loop is that the missile will automatically react to a wind gust, thrust misalignment or disturbance due to a rigging error. With antitank missiles there is always the danger of hitting the ground. When a position gyro is used in the pitch plane the missile is usually launched at a set elevation with the gyro uncaged at an angle to the missile framework so that it will fly in a horizontal plane with a small permanent incidence to overcome gravity. Without the gyro, gravity compensation still has to be obtained. Any error in biasing the elevator servos results in the missile accelerating upwards or downwards and this calls for more skill from the operator if he is to bring the missile back on course.

A velocity control system is suitable for systems designed to hit stationary or slow moving targets where small adjustments only are required by the operator. LOS systems designed to hit fast crossing targets require the missile to execute a curved trajectory and a directional autopilot would merely hinder the operator in trying to keep the missile in the LOS.

A block diagram of a velocity control autopilot[4] is shown in Figure 1.

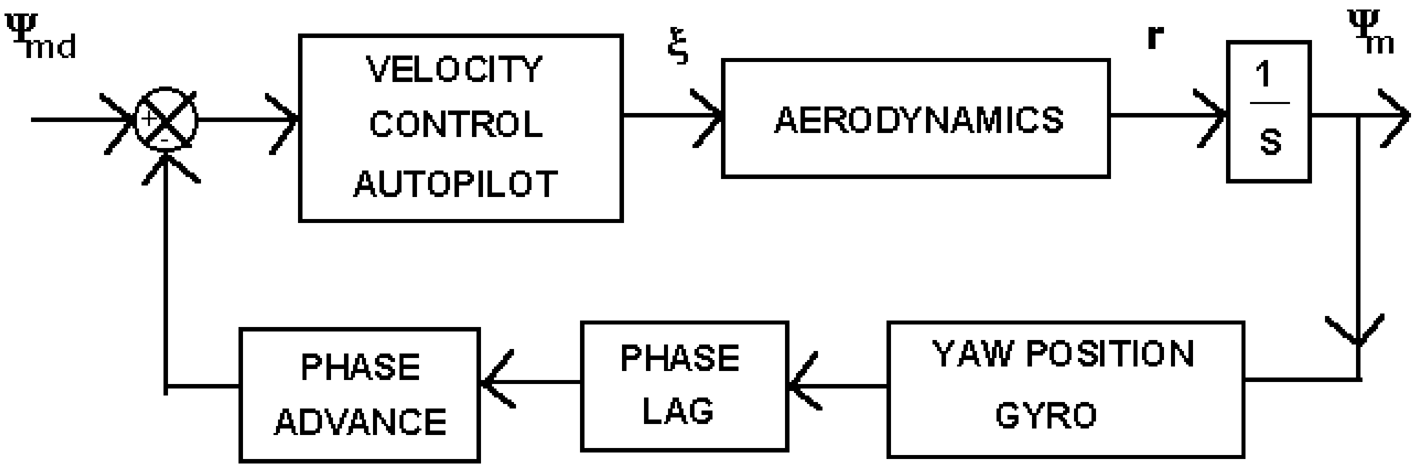


Figure 1. A velocity control autopilot.

Here r is rudder, ξ is body rate, Ψm is body direction and Ψmd is the desired body direction.

3. Fuzzy Controller and Membership Functions Description

Fuzzy controllers are described by means of some linguistic control rules set. One of the most popular type of rules is used in the present investigation:

$$IF\ x_1\ is\ L_j\ and\ x_2\ is\ L_k\ and\ ...\ x_p\ is\ L_q\ THEN\ control\ action\ is\ u_i$$

where i is number of control rule, $i=1\div N$, N is number of rules, L_j, L_k,...,L_q are linguistic values of the controllers' input variables x_1,...,x_p respectively and ui is crisp control action for i-th rule.

The crisp value u is a result of composition of all fuzzy rules with some compositional rule of inference and defuzzification.[3,10] Usually, max-min composition and centre of gravity defuzzification are used. In this case the crisp value is obtained as follows:

$$u = \frac{\sum_{i=1}^{N} w_i u_i}{\sum_{i=1}^{N} w_i}, \text{ where } w_r = \min_{i=1}^{p} \mu_{L_k}\left(x_i\right), r = 1 \div N$$

Here μ denotes the membership grade of corresponding inputs of controller (x_1,...,x_p) with respect to given linguistic values L_j,...,L_q and w_i is the degree in which *i*-th rule influences controller output.

The following parametrized functions for increasing and decreasing parts of membership functions are proposed[2]:

$$\mu_I(x) = \frac{(1-\beta)^{\alpha-1}(x-a)^{\alpha}}{(1-\beta)^{\alpha-1}(x-a)^{\alpha} + \beta^{\alpha-1}(c-x)^{\alpha}}, x \in [a,c]$$

$$\mu_D(x) = \frac{(1-\beta)^{\alpha-1}(b-x)^{\alpha}}{(1-\beta)^{\alpha-1}(b-x)^{\alpha} + \beta^{\alpha-1}(x-c)^{\alpha}}, x \in [c,b]$$

where α is parameter defining the function slope and β is the inflection point of the curve. Parameters *a*, *b* and *c* define the support (definition interval boundaries) of the function. The indexes *I* and *D* denote the increasing and decreasing function respectively. The influence of these two parameters on the shape of the membership function can be seen on Figure 2.

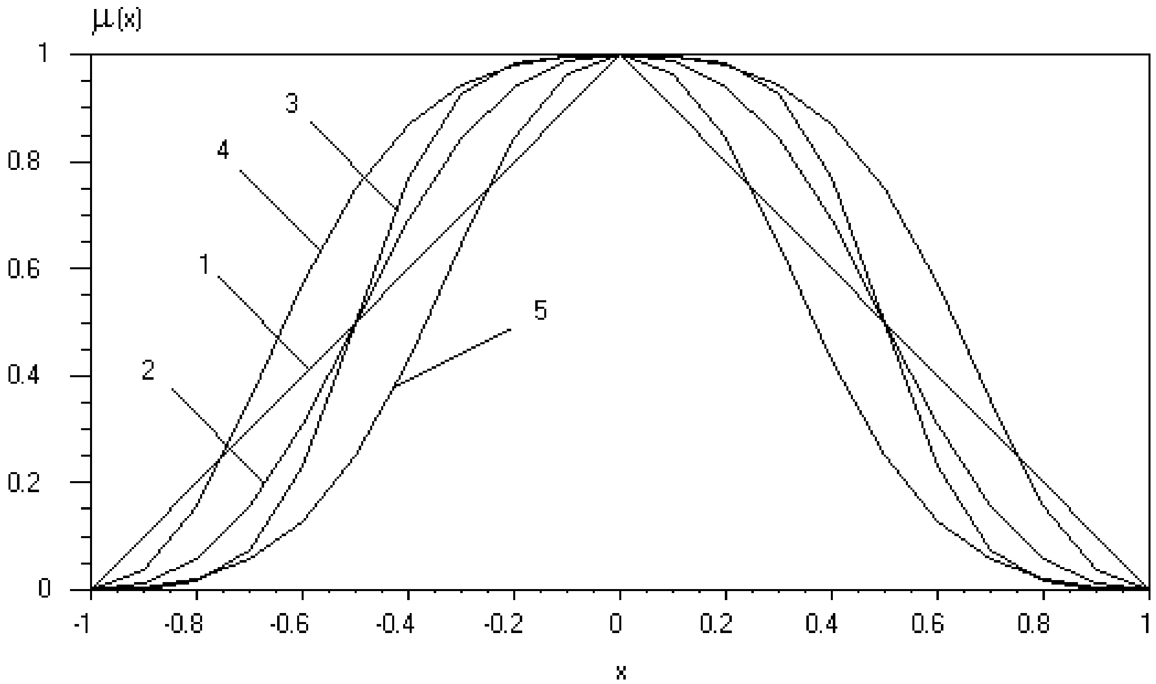


Figure 2. Influence of α and β parameters on the shape of membership function.
curve 1 - α =1; curve 2 - α =2, β =0.5; curve 3 - α =3, β =0.5;
curve 4 - α =2, β =0.25; curve 5 - α =2 , β =0.75.

4. Simulation Results and Discussion

The object under control is the missile (Figure 1) described by the aerodynamics with the following transfer function:

$$\frac{r}{\xi} = \frac{(T_i s + 1) k_a}{s^2/\omega_{nw}^2 + s\, 2\mu_w/\omega_{nw} + 1}$$

The parameter values are as follows:

$$T_i = 0.043 \quad k_a = 20 \quad \omega_{nw}^2 = \frac{1}{0.01} \quad 2\mu_w = 0.3\omega_{nw}$$

The controlled variable is body direction Ψ *m*. It has to follow the desired body direction Ψ *md*. The velocity control autopilot incorporates the operator's skills in the form of fuzzy control rules, described in the previous section.

The control action ξ *(t)=u(t)* is calculated according to the fuzzy rules table (Table 1). The fuzzy controller inputs are the error *e(t)* and its change *de(t)* (*x1=e(t)*, *x2=de(t)*), i.e.:

$$e(t) = \Psi_{md}(t) - \Psi_m(t) \quad de(t) = e(t + \Delta t) - e(t)$$

Here Δ *t=0.001* is the integration step. The supports of the corresponding fuzzy values of the two input linguistic variables (*e* and *de*) are given in the Table 2 and Table 3 respectively.

Table 1. Fuzzy controller rules.

de(t)	**e(t)**		
	L1	**L2**	**L3**
L1	0.0	+0.005	+0.01
L2	-0.005	0.0	+0.005
L3	-0.01	-0.005	0.0

Table 2. Membership functions supports for the error *e*.

e(t)	**a**	**b**	**c**
L1	-0.05	0.0	-0.05
L2	-0.05	+0.05	0.0
L3	0.0	+0.05	+0.05

Table 3. Membership functions supports for the change of error *de*.

de(t)	**a**	**b**	**c**
L1	-0.0005	0.0	-0.0005
L2	-0.0005	+0.0005	0.0

L3	*0.0*	*+0.0005*	*+0.0005*

Simulation results with different values of the parameters α and β of fuzzy controller membership functions are shown on the Figure 3, 4 and 5.

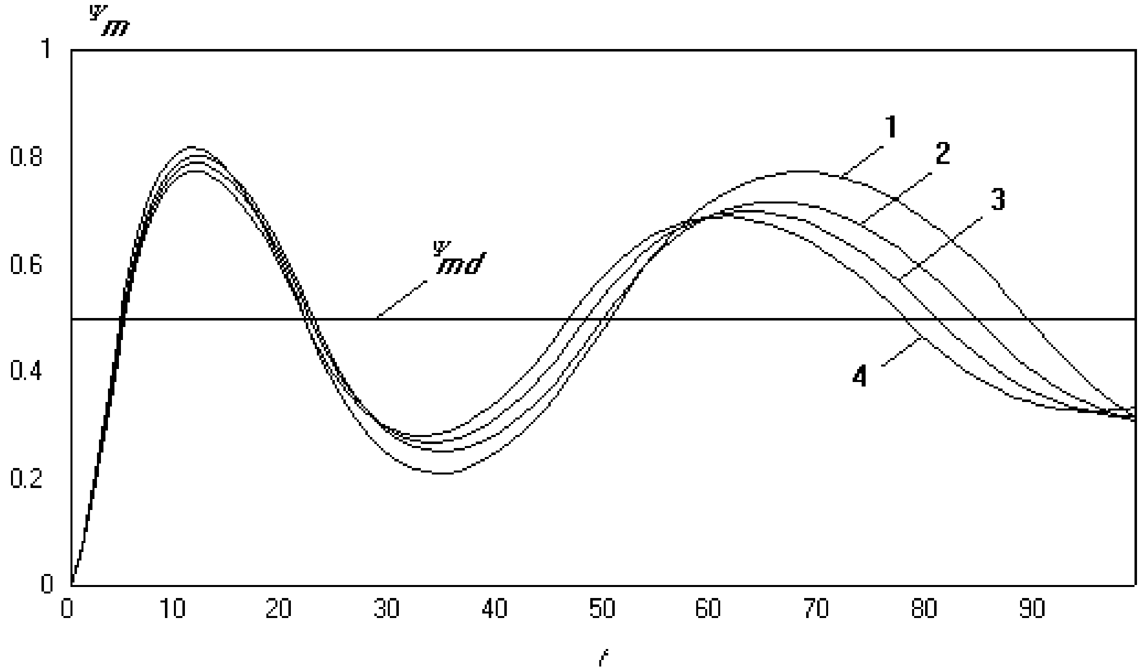


Figure 3. Simulation results with β =*0.25* and different values of α (time duration 1s).
curve 1 - α =*1*; curve 2 - α =*2*; curve 3 -α =*3*; curve 4 - α =*4*.

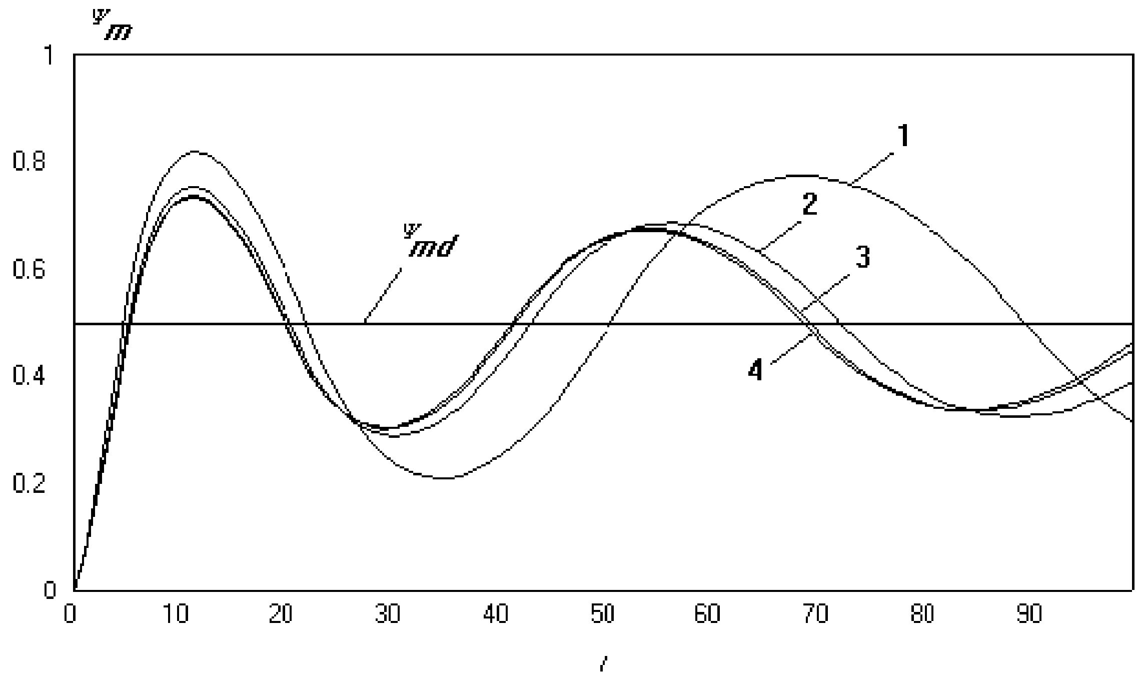


Figure 4. Simulation results with β =*0.5* and different values of α (time duration 1s).
curve 1 - α =*1*; curve 2 - α =*2*; curve 3 -α =*3*; curve 4 - α =*4*.

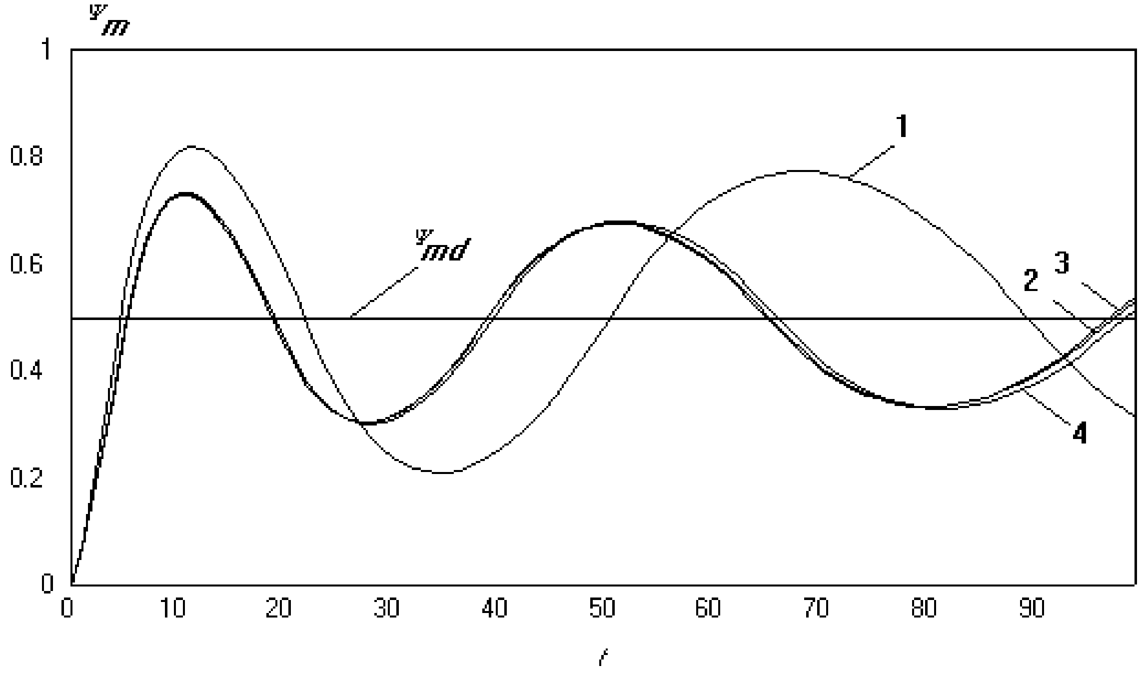


Figure 5. Simulation results with β *=0.75* and different values of α (time duration 1s)..
curve 1 - α *=1*; curve 2 - α *=2*; curve 3 -α *=3*; curve 4 - α *=4*.

As can be seen from the above figures, fuzzy control autopilot with bell-shaped membership functions decreases the amplitude and period of fluctuations around the set point in comparison with the triangular-shaped membership functions in all the cases of parameters α and β . With the increase of parameter β (inflection point) the period of fluctuations decreases. With increasing of parameter α in the cases β *=0.25* and β *=0.5* (figures 3 and 4) the period and amplitude of fluctuations decrease, but for β *=0.75* there is no significant difference in simulation curves (Figure 5).

5. Conclusions

In the present paper the influence of the two parameters defining the linguistic variables membership functions shape on the dynamical behaviour of fuzzy control autopilot for missile is investigated. The simulation results led to the conclusion that the bell-shaped membership functions give better results in comparison with the triangular once.

Acknowledgement

The work on this paper was supported by the Bulgarian Academy of Sciences, project "Development of Moving Platform simulator".

References:

1. Baglio, S., Fortuna, L., Graciani, S. and Muscato , G., "Membership Function Shape and the Dynamical Behaviour of a Fuzzy System," *First European Congress on Fuzzy and Intelligent Technologies EUFIT'93* (Sept. 7-10, 1993, Aachen, Germany), 645-650.
2. Dombi, J., "Membership Function as an Evaluation," *Fuzzy Sets and Systems* 75 (1990), 1-21.
3. Driankov, D., Hellendoorn, H. and Reinfrank, M., "An Introduction to Fuzzy Control," (Springer-Verlag, 1993).
4. Garnell, P., East, D.J., "Guided Weapon Control Systems" (Pergamon Press, 1997).
5. Kitamura, S. and T. Kurozumi, "Extended Circle Criterion and Stability Analysis of Fuzzy Control Systems," *Proc. of the Int. Fuzzy Engineering Symp. IFES'91* (Nov. 13-15, 1991, Jokohama, Japan), vol. 2: Fuzzy Engineering toward Human Friendly Systems, 634-643.
6. Koprinkova, P., "Membership Functions Shape Influence on Fuzzy Control Actions," *The First Workshop on Fuzzy Based Expert Systems FUBEST'94* (Sept. 28-30, 1994, Sofia, Bulgaria), 116-118.
7. Koprinkova, P., "Influence Investigation of the Shape of Fuzzy Variables Membership Functions on the Control Surface

Shape of a Fuzzy Logic Controller," in Proceedings of the National Conference '*Automatics and Informatics'95*,' (Nov. 7-9, 1995, Sofia, Bulgaria), 22-25 (in Bulgarian).

8. Koprinkova, P., "Membership Functions Shape and Its Influence on the Dynamical Behaviour of Fuzzy Logic Controller," *Cybernetics and Systems: An International Journal* 31, 2 (March 2000), 161-173.
9. Koprinkova, P., "Membership Functions Shape and Its Influence on the Stability of Fuzzy Control Systems," *Cybernetics and Systems: An International Journal* (in press).
10. Lee, C.-C., "Fuzzy logic in Control Systems: Fuzzy Logic Controller - Part I and Part II," *IEEE Trans. on SMC* 20 (1990), 404-434.

PETYA KOPRINKOVA is a research assistant at the Institute of Control and System Research, Bulgarian Academy of Sciences. She holds an M.Sc. degree from the Technical University of Sofia. Ms Koprinkova has more than twenty publications in the areas of control of biotechnical systems, fuzzy logic and neural networks. Address: Acad. G. Bonchev Str., block 2, P.O.Box 79, Sofia 1113, BULGARIA.

VALENTINE PENEV Born 1960. M.Sc. (1986, Automatics and Telemechanics) from the Technical University of Tula, Russia. Ph.D. (1992, Fuzzy sets in digital control systems) from the Robotics and Control Systems Research Center at the Bulgarian Academy of Sciences. Currently Dr. Penev is Deputy Director for R&D of the Institute of Control and Systems Research at the Bulgarian Academy of Sciences. Senior Research Fellow with over forty publications in the field of intelligent technology engineering - fuzzy control, fuzzy reasoning, signal processing, and neuro-like structures. Software realization of a flight simulator and the simulator "Fagot" for ATGMs under Windows95, Windows 3.1x and DOS for PC. Address for correspondence: Institute of Control and Systems Research, Bulgarian Academy of Sciences, Acad.G.Bonchev Str., Bl.2, Sofia 1113, Bulgaria. E-mail: valpenev@iusi.bas.bg

BACK TO TOP

Dynamical Behavior of Fuzzy Logic Based Velocity Control Autopilot with Respect to Changes in Linguistic Variables Membership Functions Shape

Petya Koprinkova and Valentine Penev

Keywords: velocity control autopilot, fuzzy controller, membership function

Abstract: In the present paper the influence of membership functions shape on the dynamics of fuzzy logic based autopilot controller is investigated and discussed. Bell-shaped membership functions with two parameters are discussed.

Authors: **Nikolay Vraikov, Alexi Naidenov**
Title: **The Computer Aided Exercise - An Alternative of the Conventional Exercises in the Armed Forces**
Year of issuance: **1999**
Issue: **Information & Security. Volume 3, 1999**
Hard copy: **ISSN 1311-1493**

THE COMPUTER AIDED EXERCISE – AN ALTERNATIVE OF THE CONVENTIONAL EXERCISES IN THE ARMED FORCES

Nikolay VRAIKOV and Alexi NAIDENOV

Designing and building simulation systems is a dynamically developing area, in which, for relatively short time and under definite conditions, good results can be achieved. Our army has gained significant experience in the development and use of various simulation systems. The participation in a great number of seminars, presentations and computer aided exercises gives us the opportunity to analyze the achievements in the area of development and use of simulation systems in the education and training process of the staffs at various levels.

What we should start with when discussing the problem of computer aided exercises (CAX) is to disintegrate the whole, i.e. the simulation, to define its components and based on that, to find the place of the CAX in it.

Thus the whole—the simulation–is a process of recreation and imitation of the reality (in complex systems) with a definite precision, using several interrelated and interdependent objective and/or abstract (mathematical) models, working as an integrated whole, subject to a common concept and plan. Therefore, the characteristic features of the simulation are the following:

- the simulation is not a fixed process, not a state already reached but a strictly defined process of imitating a specific object, its characteristics and way of functioning;
- existence of a number of mathematical models, different and yet compatible, complementing each other and working under a common definition and medium;
- each of these models solves a strictly specific problem, which is a part of the general re-creation of the reality.

Talking about simulation, we must have in mind that according to their technical qualities and potential capabilities we can differentiate three basic simulation methods:

1. Live simulation;
2. Virtual simulation;
3. Constructive simulation.

Under live simulation we understand training or exercises with real troops and real combat equipment on the terrain and in conditions as close as possible to the real ones.

Under virtual simulation we understand instruction and training in an artificially created environment, where through simulators used for the actions of the operator or the crew, the behavior of the entire system is represented. A primary task here is to create as genuine as possible imitation of the original system and its behavior. Two basic levels can be differentiated with the virtual simulation:

- *Level I – These are the so called individual training simulators*. Here we do not mention the interacting computer audio-visual means of training and visualization, since they rather mark the transition from the purely applied use of computer technologies to the simulation. For us here, of a specific interest are the simulators. They are used for formation and maintenance of definite habits and skills of the individual soldier (commander). The trainers are basically related to the use of definite weapon systems and means. In other words, the realization of the aims and tasks at the first level is achieved through the so called simulators. Examples in this respect are: movement and flying simulators; shooting simulators; duel fighting simulators, etc.

- Level II – *The so called higher class simulators used for tactical training of crews, commanders and staffs (Tactical training*). They are used to confirm and develop the concepts of conducting combat actions from crew level to company level, in a realistic and challenging combat which is artificially created. The purpose of using them is to put together the tactical actions of crew, platoon and company.

The next simulation method is the constructive simulation. Under constructive simulation we understand the application of simulation models using instruments for analysis, which are capable of assessing and solving the consequences from the decisions made. Here, the acceptance of the system as a whole is always determined by the likelihood of the simulation results. In the constructive simulation, two levels can be distinguished, too:

- Level I – *Computer Aided Exercises (Command and Staff Training)* – The most modern and effective method of training of commanders and staffs at a battalion level and above. They are aimed at a formation and keeping up definite habits with commanders and staffs related to control of subordinates and also at optimization of staff procedures and functions.

- Level II – *Expert systems* – Meant mainly for solving defense planning problems and strategic analyses. The simulation systems of that level support the decision making process in the area of military planning, military doctrine development and analysis of interoperability issues. They are also widely applied in the preparation of comparative analyses when making alternative decisions (different in number, structure and type of armed forces), interaction among the various subsystems (sub-components) of the armed forces and operational strategies.

As basic requirements for the development of the simulation systems used in CAXs the following can be defined:

- the software should be interoperable with the equipment and information systems used for forces control;
- the simulation systems should be capable of configuring depending on the user’s requirements, composition and level of the participants and in compliance with the exercise objectives;
- an interface, corresponding to the international standards should be used, which will allow for a common use of a number of simulation systems with the purpose of conducting comprehensive multinational CAXs;
- centralized coordination in the development and the use of simulation systems for CAXs;
- the simulation systems should be in a didactic, logic and technological uniformity.
- all simulation systems should use a common basic data-bases (for the area, vulnerability, weapon systems capacities, etc.)

Having thus defined, although in most general terms, the place of the CAX, we will further try to define its nature and outline its basic characteristics.

The Computer Aided Exercise is an interactive and non-traditional method of training, based on the application of a wide range of computer mathematical models, providing a true representation of the combat actions and offering opportunities for making certain deductions out of which a conclusion about the eventual subsequent behavior of the trainees in a real environment could be made. In other words, it can be looked upon as a super-modern method of training based on the use of modern information technologies and microprocessor equipment and the application of animation and simulation techniques on the background of an adequate visual and acoustic image.

Before moving on to the functional and technological structure of the CAX, let us focus on its advantages. In general terms they can be brought to the following:

1. It is a cheap and highly effective alternative of the conventional types of exercises, shortening considerably the time necessary for achieving the same results through conventional exercises. The interactive simulation systems used in the CAX staff exercises allow for a fast and quality planning, assessment, decision making, working out of the plan and tactics of actions in definite pre-set initial conditions. In that way, as a result of playing out a number of situations at a minimum outlay of material and human resources, the actions of various institutions, units and individuals working together to a common end are coordinated.

2. Environmental protection – The opportunity to play out various moves, simulated through mathematical models excludes the likelihood of polluting and damaging the natural environment due to discharging harmful substances from the machines. In that way, considerable funds needed for its recreation are saved.

3. Decreased risk situations – The opportunity for playing out to a degree of perfection various moves, simulated through computer models decreases the dangers when applying to practice what has been learned, since no wrong move can lead to injuries or eventual loss of a human life.

4. Saving of time and giving opportunities for multiple repetitions and play outs – Judging from international experience the training of staffs and commanders at various levels through CAXs saves up to 60 % of the time needed for acquiring the same knowledge and skills in practical training. Besides, through the simulation systems it is possible to play out repeatedly various scenarios of a certain leader's decision and also to react quickly and timely to emergency situations, i.e., to receive timely information about an expected turn in introducing the second echelon, air assault disembarkation into a certain area, etc. All that allows for a choice and working out of a more efficient training strategy.

5. Economy of resources – That feature has always been topical but in the current poor economical indices of our country, its significance is even greater. If we compare for example the funds, which will be needed for conducting an exercise with troops on the terrain to those needed when using the simulation system of CAXs, it will become clear that they are incomparable. Thus, for example, to conduct a brigade level CAX approximately DM 7000 will be needed, while an exercise achieving the same results but conducted through deployment of equipment on the terrain will already cost DM 240 000.

6. Optimization of the decision making process – The CAXs offer opportunities for studying the process and techniques of decision making. Based on the statistical information obtained, it is possible to enter repeated corrections to improve them.

7. Improvement of the communications and structures of the information flows, running among the main officials from the particular body of authority.

8. Raising of the quality of the exercise – The teams preparing and conducting CAXs are usually composed of experts with the best training. Furthermore, the simulation system is absolutely unbiased, so that classes of a high quality are offered in compliance with the preparedness level and specific physical and psychological state of the trainers and trainees.

9. Increased sense of one's own worth and motivation – The opportunity for the trainee to achieve independently

and individually the particular objectives and follow the results of his or her decisions during the exercise leads to an increased sense of their worth and motivation unlike the conventional frontal communication.

10. Orientation to the future – As a result of the use of the most modern equipment and information technologies, the volume of knowledge of the capabilities of the simulation models and CAXs is strongly increased. The simulation models, with their constant updating and improvement allow for a complete application of the latest achievements in the area of the modern interactive methods of instruction, training and fitting together of the staffs. Furthermore, they make it possible to conduct CAXs from the staffs' permanent locations and from their wartime command posts, the so called distributed CAXs. Distributed CAXs are the next step and level of using this instrument in staff training.

Specifically in the Bulgarian army, the existing simulation system for CAXs was put into operation as early as 1996. It is intended for conducting Command Post CAXs at a brigade staff level. This system is open and allows for:

- perfection of the existing models, as well as inclusion of new ones aimed at achieving a higher degree of reliability in controlling the combat actions. It is compatible with the local information computer network of the staffs trained and in future it is expected to work in a regional regime;
- all trainees from the leadership bodies are to fulfil in full their functional duties on decision making, planning the operation (battle), organizing the combat actions, ensuring a firm and continuous control, maintaining the interaction and organization of the overall support in compliance with the generally accepted techniques;
- carrying out staff training and uni-/multistage exercises of the tactical and operational staffs, without or with participation of troops;
- many-sidedness of the exercise when decision making and control of the forces takes place in the conditions of a modeled and close to reality clash with the enemy;
- assessment of the terrain and organization of the interaction of the staffs trained using a geographical information system for terrain analysis with a specific purpose;
- accumulation of certain knowledge and skills for situation assessment, decision making and forces control;
- collection, storage, processing, display and dissemination of information about the situation with all its elements;
- collection of information necessary for exercise analysis;
- training of the staffs at training centers specifically established for that purpose and, in future, also at permanently located posts;
- storing the information and the versions of the decisions made and played out.

As we already mentioned, the CAXs are a form of staff training, in the course of which simulation systems are used, providing a true presentation of combat actions. They are developed as interactive computer-based games and are intended for training of officers and staffs for fulfillment of their wartime functions.

Depending on the training aims set, tasks to be solved, trainees composition, degree of detail in forces representation, weapon systems, terrain and impact of other factors, the CAXs are subdivided into three types: group, detailed and general.

The *group exercises* have a limited number of participants and hierarchy levels. With them, a small number of problems are worked out which reflect one episode of the combat actions using general simulation model. The modeling is fast and allows for focusing on the exercise objectives.

The *general exercises* have wider objectives and participating in them are officials, whose functional duties are related to the achievement of the objectives. With those exercises the individual episodes of the combat actions are not simulated in detail but they are based on more general principal positions. The functioning of the staff trained is in real time which can also be made shorter.

Detailed exercises are those exercises with which complex objectives are set. Participating in them are a large number of organizational entities and in their modeling lower organizational levels and a larger number of factors affecting the combat actions are taken into account. With them, a system of models of the Services, Branches and special forces, and the types of support, is used. The exercise runs in real time.

Each CAX is characterized by definite technological and functional structures.

The technological structure is intended for data collection, display, dissemination and processing, and for formation and presentation of the modeling results. It ensures the dynamics of the exercise concept and makes the situation look real. Its main components are *the simulation system of combat actions modeling, the computers themselves, the technical equipment and the communications infrastructure.*

The simulation system for combat actions modeling allows for simulation of various aspects and functions of the combat actions, receives the input data for the modeling, guides the events from the operations (battle) dynamics and figures out its results.

It consists of the following systems:

- a model of the all-arms operation (battle). It provides a simulation of the combat actions of the battalions, brigades and corpses, making a complex accounting of their resources. It allows for modeling of the effect from using conventional and other types of ground-based weapons. It allows for playing out all characteristic actions of the Land Forces on the background of definite geographical system, modeling completely the terrain relief and the operation of a large number of different types of weapons;

- a model of the Air Force – It simulates the aspects of the combat actions with conventional forces and equipment of the aviation and the air defence. This is a three-dimensional and spatial model of the actions during daytime and nighttime, in simple and complicated meteorological conditions. Participating in the model are the airbases, the missile complexes of the air defence of all classes and types, and the components of the radar system. It gives a detailed account of the armaments and resources of ammunitions and POL;

- a model of the Navy – It simulates the spectrum of actions and factors of maritime battles, including convoying and transfer of forces;

- model of reconnaissance and electronic warfare – It simulates the process of reconnaissance of the enemy's sites with the full complex of bodies and techniques of actions, electronic jamming of the radar and communications systems, counteraction against the enemy's reconnaissance equipment and the radio-electronic protection of the friendly equipment for control provision;

- model of the operational and combat support and model of the logistic support - It simulates the systems for operational, combat and logistic support;

- model of NBC – It simulates a creation of a complicated radiation and chemical situation;

- model of the control system – It simulates the command, control and communications (C3) system.

Within the models the users work through a common users interface. Through it, the initial data are entered and the results from the entire modeling process are received.

The technical equipment for conducting CPX CAXs is usually set up into specialized training centers.

The computers, on which the models work and on which the communications system is based, are various types and with various capacities. There are two basic types of computers:

- workstations;

- servers.

The workstations can be:

- processor – not lower than 1486, Pentium;

- clock frequency – not less than 300 MHz;

- RAM – not less than 32 MB;

- hard disk volume – not less than 5 GB;

- monitor – SVGA;

Servers:

- processor – RISC;

- operational memory – expandable up to 640 MB;

- disk space – built in 12-16 GB, maximum – 200 GB;

- output – not less than 100 SPEC int 92;

For the local computer network:

- IEEE 802.3/Ethernet;

- 10 BASE – T;

The local computer network itself includes three types of workstations:

- for work with verbal and table battle documents (operational directives, battle orders, battle instructions, battle reports, information reports, briefings, etc.);

- for work with operational information, using topographic basis;

- for linking with the models for combat actions.

These workstations are the minimum necessary for conducting CAXs.

For UPS:

- for server – not less than 1200 Bt;

- for workstations – not less than 600 Bt

The simulation system which is used for conducting CAXs works with the following system software:

- for the work stations - Windows 95;

- for the server - UNIX OC for RISC, Windows NT 4.0;

- for the local computer network - Windows NT 4.0, TCP/IP;

- ORACLE database management system.

The applications for data input and output in the database are built with the following computer languages:

- Technological development medium DELFI 2;

- Technological development medium Borland C++4.0, 5.0]

- Technological products from ORACLE database management system.

The communications in the CAXs insures the distribution of the processed information in the preparation and course of the exercise. It is the communications and the computer interface that provide the conduct of distributed CAXs, which means to use the simulation models from remote locations, where the exercise participants are (the places of permanent location, wartime command posts, special training centers, etc.) and exchange of video and verbal information.

Figure 1 presents organizational structure and information flows in a CAX. The functional structure of a computer aided exercise is represented on figure 2.

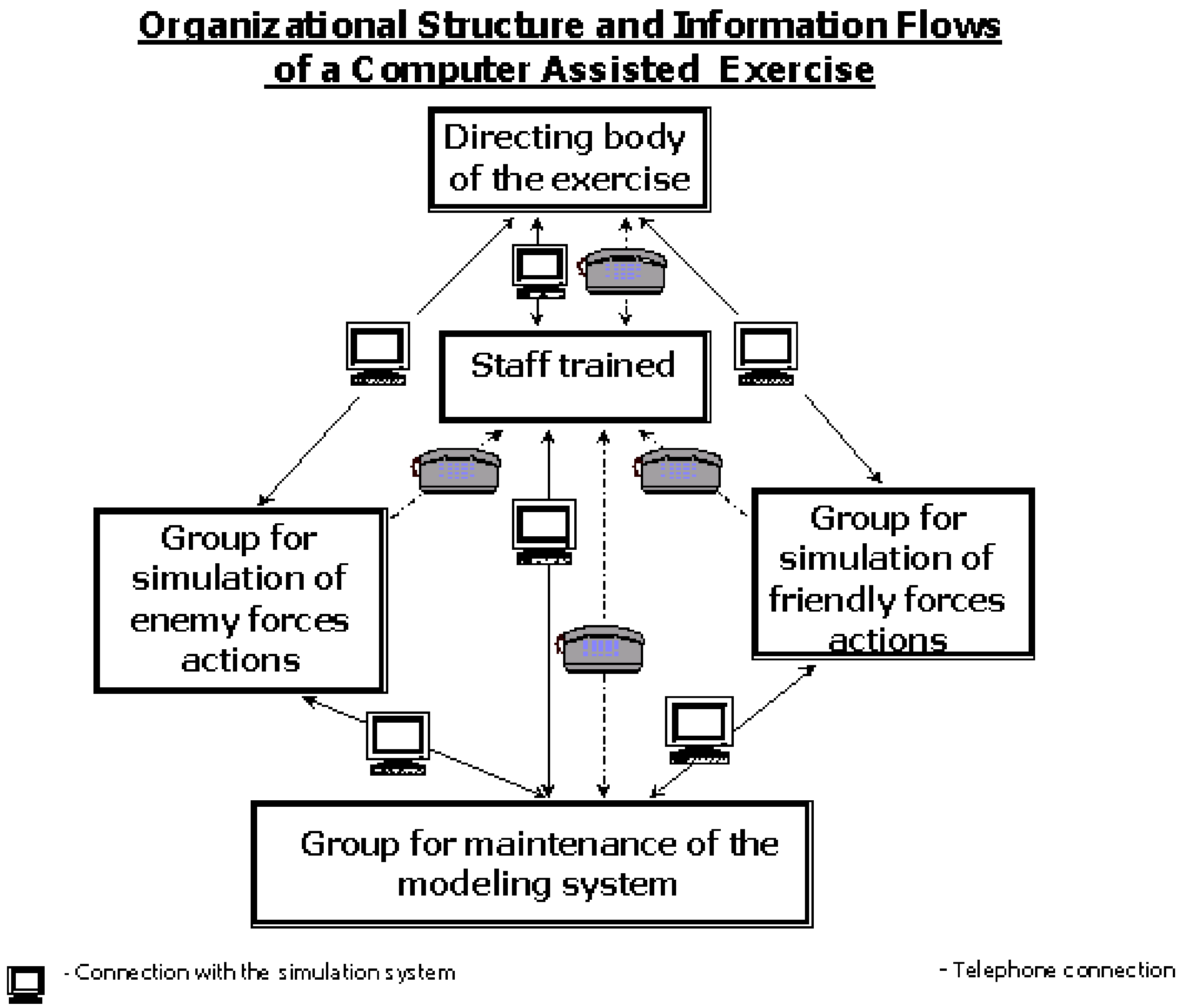


Figure 1. Organizational structure and information flows in a CAX.

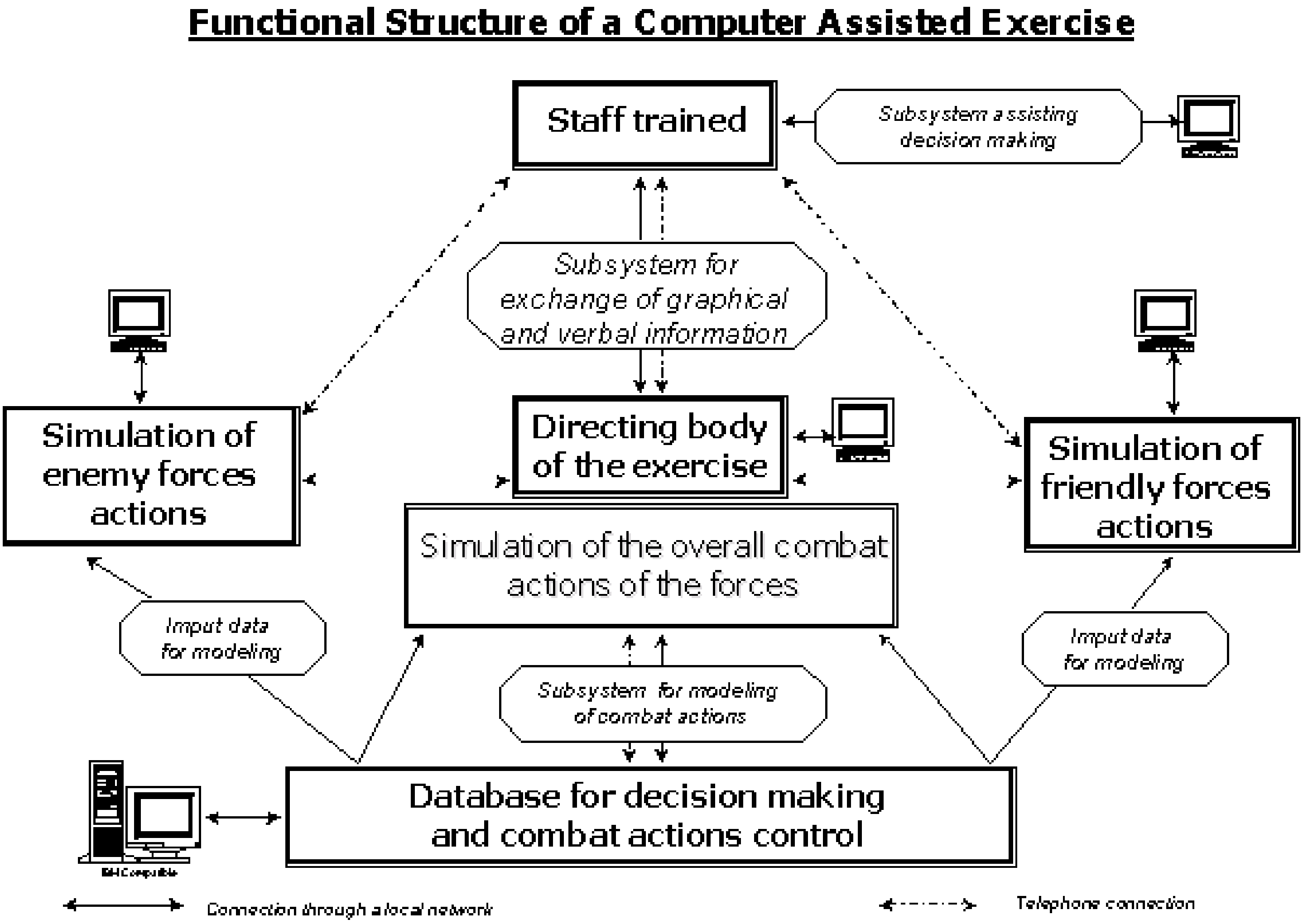


Figure 2. Functional structure of a CAX

As an organization the participants in the preparation and conduct of CPXs are divided into a directing staff, a trained staff, a group maintaining the modeling system and a supporting group.

The directing body is responsible for the preparation and conduct of the exercise and achievement of the exercise objectives. It can have the following structure: director, chief of staff and directing staff.

The director of the exercise is responsible for the timely preparation and conduct of the exercise at a high organizational and methodological level. He is determining the theme, the training objectives and problems, the area of the exercise and the time limits for the preparation of the documents.

The chiefs of staff of the directing body is to organize the staff's activity and control the development of the documents necessary for the exercise conduct, organize the work of the staff during the exercise.

The directing staff is a basic body in the preparation and conduct of the exercise. It is to develop all documents related to the exercise. It includes a group for interaction, a group for simulation of the friendly forces actions, a group for simulation of the enemy forces actions and a group for control and analysis.

The group for interaction simulates the actions of the superior staff versus the staff trained and the staffs of its neighbors or realize the functions, which are not worked out during the exercise.

- The group for simulation of the friendly forces actions simulates the actions of the subordinates to the staff trained, staffs or forces, depending on whether the exercise is a two-stage or a multi-stage one.

- The group for simulation of the enemy forces actions develops the concept of the counteracting party, models the actions on the ground, water and air in the course of the operation (battle), thus counteracting the decisions made by the trainees.
- The group for control and analysis controls the course of the exercise, ensures the achievement of the objectives and analyzes the activity of the staff trained.

The staff trained is in its real personnel composition, all officials performing their functional duties.

The maintenance group prepares the model, observes the modeling process, takes care of the stable work of the overall system, keeps track of the network and communications equipment and maintains the operating data exchange system.

The supporting group provides for the life support of the exercise participants.

The organization itself and the conduct of the exercise determine the run of several basic information flows.

The first one includes the whole information generated and exchanged through the simulation model. Included in this information cycle are: the exercise directing staff, the group simulating the friendly forces actions, the group simulating the enemy actions and the modeling group. The information running here is graphical and verbal. Through it the clash of the two warring parties and the course of the combat actions. This information cycle insures a reliable control and, when necessary, interference in the course of the exercise by the modeling group. The medium for this information flow is the network built, which connects the servers, the work stations and the equipment for the output of graphical and verbal information.

The second basic information flow is the flow running between the directing staff, the training staff and its subordinate units. The realization of the information exchange is accomplished through the communication channels established by means of the available communications means - telephones, radio stations, etc. It provides for the exchange of current information above all between the staff being trained and their subordinates in connection with the changed situation. The reliable functioning of this information flow determines the quality of the exercise conduct. All changes of the situation which are displayed at the work stations using the available communications equipment are reported by the subordinates to the commander of the staff trained, since he is unable to follow the change of the situation through a workstation. He receives information about the course and development of the combat actions from his subordinate commanders' reports only, as it is in a real situation. The functioning of this information flow provides for reporting of the situation by the commander of the staff trained to the directing staff, for assigning the tasks to the subordinates and receiving directions from the directing staff.

The third basic information flow during the conduct of the CAX insures the exchange of graphical and verbal documents between the exercise directing staff and the staff trained. This information channel is used for sending battle orders, reports, instructions and visual graphical information. It is realized through the local network and computers with special designation.

In conclusion, we must emphasize that the solutions of the issues related to the use of simulation systems in the training process of the Bulgarian Armed Forces in the next several years will be one of the main directions in the army's activities. A basis for that is the Military Doctrine of the Republic of Bulgaria and the Plan for Organizational Development of the Armed Forces till 2004.

The subdivision of the Armed Forces according to their designation into Rapid Reaction Forces, Defence Troops, Territorial Defence Troops and Reserve Forces, and the new functions of the Armed Forces in accordance with the

Military Doctrine of the Republic of Bulgaria as deterring and defensive functions, peace-support functions, humanitarian and rescue functions, international and social functions necessitate intensification of staff preparation, implementation of new forms and methods of training. This has been reflected in Article 95 of the Military Doctrine – "In the forces and staff training, alongside the conventional methods, simulator and computer supported forms of training are also used…"

Theoretically, the issue of using simulation systems in the Bulgarian Armed Forces is further developed in Plan 2004, where it is said that "the methods of staff training will be perfected for acceptance of the modular principle of preparation and conduct of simulation exercises and exercises conducted according to the standards and procedures established in the armed forces of the NATO member states and in order to achieve the interoperability objectives and adapt the standardization documents and staff procedures to our conditions, the relative share of the Computer Aided Exercises will be increased…"

All that determines as a primary task to be realized by the army during the next coming years the mastering of new and more effective forms of training of the staffs and forces based on effective use of the simulation systems.

References:

1. Arfeev, N., "Using Computer Technologies in Troops Training," *Military Thought (Voennaya myisl)* 12 (1988). – in Russian.
2. Alexi Naidenov, Nikolay Vraikov, "The Computer Supported Exercise – A Modern Way of Knowledge Acquisition," *Military Journal* 105, 1 (1998), 48-52. - in Bulgarian.
3. Valentine Penev, "Simulation Modeling in Military Affairs: Status and Perspectives," *Information & Security* 1, 1 (Summer 1998), 91-102.
4. *Modeling and Analysis of Conventional Defense in Europe*, ed. R. Huber (Plenum Press, 1996).
5. Huber, R., L. Jones, and R. Jones, *Military strategy and tactics computer modeling of land war problems* (Plenum Press, 1985).
6. User’s Manual for the Warrior Preparation Center Ground War Simulation Model. WPC (March 1991).

NIKOLAY VRAIKOV, Ph.D., graduated from "V. Levski" Army Academy in 1985. For several years he has served at a number of garrisons of the Bulgarian Army, after which in 1991 he received doctoral degree in mathematical processing of empirical data. In the period of 1991-1994 he worked as a research associate at "G.S. Rakovski" War College. In 1996 he completed the full course of education of "G.S. Rakovski" War College in the specialty of Automation in Management. Then for one year he worked as a senior assistant at "Automation Department" of the General Staff where he was in charge of the computer systems used by the Navy and the conduct of computer assisted exercises. Since 1996 he is working as a senior assistant at "Exercises, Training and War Games" Department at the Operations Directorate of the General Staff on the issues of simulation systems for CAX, as well as on problems of collection and dissemination of their results, inference and lessons learned.

ALEXI NAIDENOV is an officer, chief of "Exercises, Training and War games" Department at the Operations Directorate of the General Staff. In 1977 he graduated from "V. Levski" Army Academy. Then he served in a number of Land Forces units. In 1986 he graduated from "Frunze" War College in Russia. After that he consecutively took positions at various staffs – Corps HQ, Land Forces Headquarters and General Staff of the Bulgarian Armed Forces. Since 1993 he is involved in designing and implementing the policy of using simulation systems in staff and force training.

BACK TO TOP

Information & Security. An International Journal
e-mail: infosec@mbox.digsys.bg

Author: **Juliana Karakaneva**
Title: **The Modular Simulation Language (MODSIM) - A Powerful Tool for Computer Simulation**
Year of issuance: **1999**
Issue: **Information & Security. Volume 3, 1999**
Hard copy: **ISSN 1311-1493**

THE MODULAR SIMULATION LANGUAGE (MODSIM) - A POWERFUL TOOL FOR COMPUTER SIMULATION

Juliana KARAKANEVA

Table Of Contents:

Introduction

One of the important directions in enhancing the processes of planning and management of defense and armed forces is the use of contemporary methods to support decision making. The application of these methods requires analysis of functional processes, technology, methods and means for their realization. The development of models, the research and assessment aimed at optimizing the process on different levels are achieved on the basis of comprehensive and in-depth analysis.

We develop a concept for using modeling and simulation in studying problems of command and control systems in conditions maximally close to the expected reality. So, the considered approach is applied to the warfare control system and in particular to the analysis of different aspects of these systems. In many cases the purpose of such models is to explore and analyze various sides of the real systems or subsystems, or on a lower level - of their elements.

In this article the discussion object is one aspect of modeling - the use of specialized software for simulation experiments with discrete models and, in particular, with models including random processes. These are systems in which random processes are evolving, i.e. random events changing the state of the system are appearing. Approaches and methods exist that allow to present and process complex sequence of events, such that may develop accidentally and asynchronously.

Computer simulation

Specialized programming languages are created for simulation experiments.[1,2] These languages include opportunities for representation of models (objects, attributes, actions), algorithms for events control, mechanisms of objects interoperability, servicing libraries of subroutines for statistics and for preparation of final results, etc.

There are two general categories of computer simulation: continuous simulation and discrete-event simulation. A continuous simulation describes a system using sets of equations which are solved numerically with respect to time. The continuous simulation program then steps forward by the increment of time chosen for the time step and recalculates all equations which describe the model.

In this case the attention is directed towards the discrete-events-based computer simulation which describes a system through logical relationships between its elements and where changes of state are caused in discrete points in time.

In a discrete-event simulation varying amounts of time elapse between events, but the state of system changes when one of its component parts changes state. The Modular Simulation Language MODSIM [3] takes the capabilities of discrete systems modeling languages like Simula and SIMSCRIPT II.5 and adds object-oriented programming capability and the modular construct of Modula-2. These properties make it an advanced tool for simulation, as well as for warfare models development.

The classical approach to discrete-event simulation involves processing of event sequences. For instance, in a simple "fire engagement" model the event routines might be:

- *Aircraft takes off*
- *Aircraft enters queue*
- *Aircraft attacks*
- *Air-defense detects the target*
- *Air-defense fires on the target*
- *Target is engaged (or target leaves), etc.*

The simple model consists of two simulations - simulation A1, representing an air force object (*AirForceObj*) and simulation A2, representing an air-defense object (*AirDefObj***)**. The routines describe discrete events and passage of time is handled by scheduling the next event for the object currently being manipulated.

This approach is adequate for smaller models, but in larger models it is difficult to follow or modify the logical scheme which describes the behavior of an object. In this case the process approach simplifies the larger model design by allowing many aspects of an object's behavior in a model to be described in one method. It creates classes of objects and each class describes the corresponding functional object's behavior and the simulation program can generate multiple, concurrent instances of this object. For example, the simulation program would create a new instance *AirForceObj* each time an aircraft takes off. While there would be multiple, distinct copies of the object operating simultaneously, each one could have different values of its fields to describe particular properties.

The considered process approach is supported in MODSIM. It combines object-oriented features and discrete-event simulation for development and maintenance of large models. A MODSIM simulation model defines a system through processes because this technique provides a powerful structure for describing multiple simulation problems, and provides many advantages over the direct use of discrete events.

Project creation

MODSIM program is created as a project that includes several types of modules. A mechanism is proposed for creation, modification and work with the project by sequence of options.

MODSIM supports a possibility for importing definitions and declarations between different modules. There are two major types of modules in MODSIM: *Main* modules and *Library* modules, and they are compiled separately to ease the task of development and maintenance. There are two parts of a Library Module - *definition* module, which contains declarations of types, constants, variables, procedures and methods, and which can be imported to other modules, and *implementation* module containing the implementation code of procedures and methods for objects, but which does not have to be visible outside the library module itself.

Data structures. Objects

The data is organized in object structures, manipulated and managed as compact information units. The object type contains a list of definition fields, describing the object properties. For instance, the information about an air force object (*AirForceObj)* includes:

Name (type)

Time between two takeoffs (random distribution)

Speed (AirSpeed)

Flight height (H)

Flight corridor (X1, Y1), (X2, Y2), etc.,

and respectively for an air-defense object (*AirDefObj*):

Name (type)

Number of Ammunitions

Fire range (R, r = f(H))

Number of Ammunitions in a shot (ShotForHit)

Time between shots (TimeForHit)

Probability for target detection (ProbForDet)

Probability for target engagement in a single shot (ProbForHit), etc.

During the experiment a possibility is created for interactive mode change of values in the objects characteristics realized through a service function in a special menu-module.

The modular structure of MODSIM program helps for flexible approach to the data base design. The developer may make changes to the methods realization and add new object types - inheritors (for example- helicopter, balloon, air-landing, etc.) to data structure and later will be able to recompile only those modules which have been changed. Through the language technology the data base is created as an open structure, where the new object types are added without problems.

Processes and events management

Besides possessing object-oriented programming features MODSIM is capable of building process-based discrete-event simulation models.

Time management

The use of time management is a key element of interoperability between processes and objects in simulations. MODSIM provides powerful and flexible tools for managing objects' behavior as model elements. A MODSIM object is capable of carrying on multiple, concurrent activities each of which elapses simulation time. An *activity* is what occurs in the model as time elapses. An *event* is a point in time at which the state of the model changes.

An object has methods associated with it to describe actions it can perform. ASK method is similar to the conventional procedure call, but when the ASK statement is executed, a message is sent to the object requesting it to invoke the method for an action. ASK methods are not allowed to pass any simulation time. For instance:

ASK Squad TO Add(Enemy),

ASK Enemy TO SetSpeed(EnemySpeed), etc.

Unique MODSIM capabilities for models realization and management are the TELL methods, which provide elapse of simulation time. The TELL call starts the simulation under the MODSIM simulation engine control. During an *activity*, time is elapsing and no MODSIM code is executed to cause this happen. During an *event*, MODSIM code is executing to change the model's state, but no simulation time is elapsing. So,

TELL Enemy TO GoTo(X,Y

TELL (ASK Tank Gun) TO Blink(CX,CY)

are statements, which elapse simulation time using a WAIT statement.

Interaction

One MODSIM object can have multiple TELL methods carrying on activities simultaneously with respect to simulation time. A method can perform a sequence of related concurrent actions. These actions may be punctuated by intervals during which simulation time elapses, i.e. they perform a WAIT. When a WAIT statement is encountered, MODSIM engine saves the state of the time-elapsing method, then suspends its execution until the WAIT is completed or interrupted. MODSIM then resumes execution of the time-elapsing method at the appropriate simulation time. During the WAIT, other activities may take place.

Several distinct object instances can be carrying out activities at the same point of simulation time. One object instance can have two different TELL methods carrying out activities at the same point of simulation time. One particular TELL method of an object instance can be invoked multiple times so that distinct instances of that method are each carrying out activities at the same point of simulation time.

Spatial management

MODSIM is provided with an object-oriented graphic system SIMGRAPHICS. The features distinguishing SIMGRAPHICS from other graphic system are its portability and its integration with MODSIM simulation engine. SIMGRAPHICS employs vector graphics to draw icons, charts and backgrounds, and allows simple approach to the use of interface, graphic and animation features of window systems such as Microsoft Windows and X Windows. The system supports the following functions:

- Data visualization, charting, plotting

- Animation and interactive graphics
- User interface through dialog boxes, menus, etc
- Editing graphical and user interface objects
- Recording and playback of animated graphics
- Three dimensional animation

SIMGRAPHICS objects (icons, graphics, forms, images) can be created through graphic editor SIMDRAW, integrated in MODSIM environment or through a MODSIM program and can be manipulated in the same way.

Experimentral model

A simple computer simulation program models the "fire engagement" operation in conditions close to the reality. The goal for the model is to get test results about fire engagement effectiveness for various object types - military objects. The rules governing the objects' behavior and the logical objects' interaction are presented. Objects involved in the simulation are:

AirForceGenerator	*AirForceObj*	*Explosion Obj*
AirDefenceGenerator	*AirDefenceObj*	*TypeObj*
GunGenerator	*GunObj*	
MenuGenerator	*MenuObj*	

The following steps are realized by the MODSIM project:

- Updating objects' parameters through menu-options (*MenuGenerator*) and air-force and air-defense objects initialization
- Presetting the experiment conditions - battlespace, terrain, air-force units flight corridors, air-defense units positions, etc.
- Generation and graphic presentation of objects and terrain
- Starting the simulation experiment

The scenario includes:

- Target detection in detection range with probability *ProbForDisc*
- One shot fire engagement in fire range with probability *ProbForHit*
- Attack and destroy target, while given conditions are available

The main conditions to consider are:

- Targets' speed and height
- Air-defense detection range
- Air-defense fire range
- Available ammunitions
- Terrain

The fire engagement effectiveness is calculated as

E = Number of destroyed targets / Number of arrived targets.

For a more precise model development other conditions should be taken in mind.

Conclusion

MODSIM possesses all the capabilities of object-oriented programming languages for the development of an information base and object classes, a mechanism for process and events control, tools for experimental data statistics and graphic user interface. Input/Output, simulation, animated graphics are available through standard modules. A unique property of the language is the *powerful and flexible engine for building process-based discrete-event simulation models.* The object-oriented graphic system SIMGRAPHICS contains means for data visualization, user interface, three dimensional graphics and animation.

Beyond the scope of this study remains the object-oriented framework for developing commercial quality discrete-event simulation applications SIMOBJECT (based on MODSIM). COMNET III (Communications Network Planning Tool) and SIMPROCESS III (Business Process Reengineering Tool) are complete applications, which are built on the SIMOBJECT foundation.

MODSIM with SIMGRAPHICS and SIMOBJECT is a powerful software tool, which enhances the ability to rapidly prototype real-world system. By the means of the program system there can also be created mathematical models based on Game Theory, Applied Queuing Theory, Graphs Theory, Morphological Analysis, Linear Programming, etc. The problems of planning, activity networks, project management, analysis and management decisions are solved.

The MODSIM-based models are mobile structures, which can be used together in different simulations, on different levels of compatibility. The programming language provides real-world process simulations building, in the military field, respectively, aimed at obtaining experimental and statistical data which would support the development of command and control systems, as well as the development of systems with education and training purposes.

References

1. T.J. Schriber, *Simulation Using GPSS* (New York: Wiley, 1974).
2. *MODSIM II. The Language for Object-Oriented Programming* (CACI Product Company, 1994).
3. Slavcho I. Slavchev, Juliana Karakaneva, V.L. Radeva, "Implementing the Programming Language MODSIM II for Development of Imitation Models," In *Proceedings of the 1997 Conference of the Bulgarian Army Academy* (Veliko Tarnovo, 1997).

JULIANA KARAKANEVA is born in 1953. She holds a M.Sc. degree in Automatics and Telemechanics from the Technical University of Sofia (1976), Ph.D. degree from the Technical University of Sofia (1983) with dissertation on "Methods for Logical Design of Fail-Safe Automatic Devices for Railway Transport." Research fellow at the Military Research Development Institute of General Staff of Bulgarian Armed Forces, currently - Institute for Advanced Defense Research. Main research interests and activities: design of command and control systems, computer assisted exercises, combat modeling. *E- mail*: gpavlov@md.government.bg.

BACK TO TOP

Information & Security. An International Journal
e-mail: infosec@mbox.digsys.bg

Author: **Jordan Parvanov**
Title: **Some Problems in Modeling and Simulation Area in the Bulgarian Armed Forces**
Year of issuance: **1999**
Issue: **Information & Security. Volume 3, 1999**
Hard copy: **ISSN 1311-1493**

SOME PROBLEMS IN MODELING AND SIMULATION AREA IN THE BULGARIAN ARMED FORCES

Jordan PARVANOV

Table Of Contents:

Introduction

A new trend in IT technology during the last 10 years has made modeling and simulations possible in the field of military training and education. Now, many countries all over the world create their own models and simulation systems for military and civil use. These systems may be used for building new types of weapons in order to design ships and airplanes, and, last but not least, for military personnel training and education.[1] As unbelievable as this may seem, simulations with the help of political systems models have even been used as a tool for predicting political crises – a domain in which Bulgarian society has known considerable tremors in recent years.[2] Modeling and simulations have also been used as a tool for analyzing country-specific economic situations.

In the armed forces, using techniques as modeling and simulations has gained popularity in the last few years mainly because of the cost-reducing effects they have on military personnel training and education. At the same time, the creation of models and the building of simulators require a solid scientific infrastructure with competent scientists and a skilled workforce able to create, develop and maintain them.

Worldwide, many scientific organizations, institutes and laboratories are involved in modeling and simulator building processes, as they each perform a task in the creation process of a common system of models. All these organizations fall under MoD control and its requirements for models and simulators. Today, as part of a new trend in the world, different non-profit and non-governmental organizations are also getting involved in building and developing different sorts of models providing effective control of military forces.

The essence of combat modeling

Without a doubt, there is an interoperability problem between the many modeling and simulation environments available today; a problem that is more pressing every year as the demand for simulations dealing with increasingly complex and heterogeneous systems grows. Military simulation in itself is significantly different from most "civilian" simulation applications and requires therefore some unique approaches.

Military simulations differ from other simulation applications in several aspects, including:

- The wide ranges possible in the scope and resolution of tactical battlefield scenarios,
- The large number of entities that can be involved in tactical battlefield scenarios,
- The hierarchical structure of many of the entities in tactical battlefield scenarios,
- The complex, dynamic relationships among entities in tactical battlefield scenarios,
- The "non-cooperative" entity interactions that characterize tactical combat, and
- The need to manage multiple levels of entity aggregation and fidelity.[3]

In its essence, a combat model is a common informative representation in electronic form on a graphical user interface, which portrays the status of army units. All this information is represented on different situation pictures – on water, on the ground and in the air, and in an electromagnetic or radioactive environment. Having all these situation pictures available allows a complete simulation. Most models created (although it depends on their purpose) represent two actors, conventionally named red and blue. Both these actors are given tasks by the people using the simulators and acting as commanders. Subsequently, these commanders analyze and evaluate the way in which these tasks are accomplished or remain unaccomplished in order to identify the weak aspects of the actors':

- Strategy and tactics;
- Interaction with other forces;
- Logistics;
- Intelligence;
- Radiolocational observation;
- Medical ensurance;

To make this analysis possible most of the models provide (not only on graphical user interface, but also in table form or on graphic charts) information the user can benefit from at the beginning and at the end of a simulation process.[4] This allows for the flexibility that the commanders need to follow the situation's changes and to react accordingly.

There is, however, too large a gap between the user's problem and the model description that the simulation program understands. In fact, commanding in the different simulation programs should resemble more the way a commander gives orders to his subordinate units in reality, that is in three steps: finding a platoon-like or battalion-like unit first, then giving them orders and following their accomplishments. In other words, the simulated environment should be a great deal closer to the real life environment.

Military simulation applications often require that experimental scenarios cover both a large scope and a fine resolution, a combination that can result in a large number of individual entities. Often these requirements can be traced back to the capabilities of modern tactical sensor systems, which can observe very large areas of the battlefield at unprecedented levels of detail.

A new trend in modeling is the representation of human factors such as the psychological training and the morale of the troops and pilots, which greatly affect the effectiveness of the armed forces in today's battles.

At the beginning models were started on separate machines and all users would be connected to these machines providing them with the modeling results. During the last 20 years distributed interactive modeling and simulation has become a widespread approach. Using this method, different kinds of models run on separate computers and exchange information between them concerning the combat situation changes on the battlefield.[5]

Models may exchange data and messages using a common database – in most cases these are ORACLE, INFORMIX or other databases which allow object-orienting representation of data.The idea of using the object–orienting approach was born at the begining of the nineties and is actual today.

Components of models, like platoons and battalions, are typically available in model libraries. Using a graphical user interface, the commander gives orders to subordinate units and these tasks are memorized in a database. When we use distributed interactive simulation, different models may request the database to allow them to read the current status of an object but only if the database logic permits this. The problem therefore is to build this logic as to make it meet the needs of interoperability between the different kinds of participating models. Another problem concerning distributed interactive simulation (DIS) is time synchronization between all models.

In order to control the movements and other activities of so many entities, it is necessary to make good use of the hierarchical structures of military units.[3] In battlefield scenarios of this size, it would be convenient to be able to manipulate entire divisions as single entities, defining division area boundaries and setting high-level objectives. However, a division is a complex organization and does not always act as a single entity. A more practical approach requires most combat and combat support units to be controlled at the battalion level, with special units, such as reconnaissance, being handled at company level or below. A good set of tactical simulation models must include support for modeling the processes by which military units receive orders, assess the situation, decide on a course of action, and generate orders for their subordinates.[6] As we mentioned above, some models are started on one machine and can be used as models to help commanders make decisions. Here the commander gives commands to both his subordinate units and the enemy units according to his expectation of what the actions of the latter will be. After the end of a simulation he then receives the results from the model. This helps him make decisions and build different variants of decisions. In this case, the problems concerning the database logic do not exist since the database is placed on the same computer as the starting location of the model.

The new trends in information technology in the last few years have created an opportunity to build WEB-based simulations, which allow users to interact using Internet-based networks.

The current state of modeling and simulation in the BAF

The BAF have implemented two kinds of models (developed in the Military Scientific Research Institute, now IADR – Institute for Advanced Defense Research) by a small group of which the author was a member. These models are to be used for situation estimation (to assist commanders making decisions) and for officer training. Both models are for ground forces purposes only. They include some elements of the Air Force and other forces as well, but detailed models for other services in the BAF are not included in these models.

Due to a lack of money and qualified people, Bulgaria has not had the opportunity to achieve the results some other countries have.

The very schematic representation of the environment may be seen as one of the main disadvantages in the existing BAF models. Among other gaps are:

- The lack of a detailed representation of AIR and NAVY situations;
- The lack of logistics representation;
- The non-representation of the communications and information system, which is one of the basic systems in today's operations;
- The fact that the NATO countries' graphical signs are not used, a problem that is relatively easy to solve on short notice.

All the other disadvantages, however, make us think that we have to develop a new model using the original models.

Despite all the disadvantages, methods for staff preparation and education using the existing models achieve positive results, but have its critics, too. The basis for these conclusions are the highly estimated exercises, which were conducted using these models.

Frequent structural reform on the scientific research level in the BAF have not allowed us to solve all the problems and disadvantages in the existing models. At the beginning of the new millenium the BAF do not have an organization or even a team responsible for the development and maintenance of today's models. Modeling and simulation is one of the decisive factors in Bulgaria's efforts to acquire NATO membership. They are identified as areas of great importance to which not only the US but also the European countries must pay great attention. If we had invested more time and effort, our models today would have been more competitive than they are now. But at the beginning of model utilization we met with strong resistance from the commanding staff. We spent considerable time convincing commanders of the advantages of model utilization. To be honest, the commanders were not opposed to the new models, but simply to the new developments in staff preparation simply because they all looked for the disadvantages, forgetting the advantages of what was being offered. The doubts arose often because there has not been an opportunity to compare our models with those developed in other countries.

It would be a great advantage to accept the conduct of CAX (Computer Assisted Exercises) using Bulgarian models. This tool is very useful in today's difficult economic situation . The use of CAX could change an old technology in the conduct of exercises and in the decision making area. In these areas the models play a positive role as they give commanding officers a dynamic change of situation without having to use real forces. This is another purpose and advantage of CAX. If all possible situations were present in the models they would provide for a better sense of reality and confidence.

Conclusion

At the begining of the new millenium Bulgaria has great opportunities to exchange ideas in the field of military modeling and simulation. The PfP Consortium of Defense Academies and Security Studies Institutes and the PIMS program provide such a chance.

We do not prepare specialists in our universities and academies for modeling and simulation purposes as this discipline has an insignificant place in their curricula. Many countries have universities specialized in modeling and simulation. In Bulgaria only the Technical University in Varna prepares students for the challenge of the new trends in information tehnology. Most of the people identify modeling with 3D modeling, but this is actually only a small part of a larger modeling discipline.

We have not conducted research and studies with the goal of discovering what the state of affairs is in the field of modeling and simulation in Bulgaria. This makes the task of modeling development spontaneous and without a strategy or concept. A concept has not yet been developed in this area. In most cases this is a reason to think that it is an underdeveloped and unimportant aspect of the force's preparation.

I would like to point out that conclusions like these are possible only after studies and research activities as the ones mentioned above as they provide a concrete task of solving the problems in this area. The economic situation does not allow us to solve the existing problems quickly. But the economic situation is not the only problem - there is a lack of personnel as well. Modeling and simulation require very well prepared personnel, with a long-time experience in the field, which we hope to create and develop. We have to give them the chance to develop this expertise in the Bulgarian Armed Forces as is the case in many countries all over the world.

References:

1. NATO modeling and simulation Master plan, Version1.0, 7 August 1998.
2. See for example Muller A. and D. Law, "Crisis Management and Conflict Prevention in a Historic Transition Period," *UNIDIR Newsletter* 30 (November 1995), 27-38. A sample of works implementing extensively modeling and simulation in the design of forecasting schemes includes

    - Peter Brecke, "Using Neural Networks to Find Harbingers of Violent Conflicts," in Preprints of the IFAC Conference on *Supplemental Ways for Improving International Stability* (Vienna, Austria, September 29 - October 1, 1995), 183-190.
    - Todor Tagarev, Petya Ivanova and Alfredo Moscardini, "Soft - Computing Methods for Early Warning of Violent Conflicts," in *Proceedings of the 1996 AFCEA-Europe Sofia Seminar*, (AFCEA, Sofia, 1996), 44-51.
    - Philip Schrodt and Deborah Gerner, "Using Cluster Analysis to Derive Early Warning Indicators for Political Change in the Middle East, 1979-1996," *1996 Annual Conference of the American Political Science Association* (San Francisco: 28 August-1 September 1996).
    - Todor Tagarev and Petya Ivanova, "Computational Intelligence in Multisource Data and Information Fusion," *Information & Security. An International Journal* 2, 1 (1999), 33-49.

3. A description of the conflict between heterogeneous simulations and interoperability is available at this URL: http://www.modelbenders.com/papers/heterop/heterop.html.
4. High-level Architecture Rules, Version 1.3, 20. April 1998, IEEE P1516/D1 Standard.
5. Richard M. Weathery, Annete L. Wilson, Bradford S. Canova, Ernest H. Page, Anita A. Zabek, "Advanced Distributed Simulation through the Aggregate Level Simulation Protocol," *Proceedings of the 29th International Conference on System Sciences* (Wailea, Hawaii ,3-6 January 1996), Vol.1, pp. 404-415.
6. Conceptual Model of the Mission Space, Information Paper, June 11, 1998, available at http://www.dmso.mil.

JORDAN PARVANOV is a Ph.D. student at the Institute for Advanced Defense Research, Defense Academy "G.S. Rakovsky", Sofia, Bulgaria. He is 1990 graduate of the Bulgarian artillery and air defense academy and holds an M.Sc. in automated control systems. Mr. Parvanov is army officer with the rank of major. His research interests are in the area of modeling and simulation. *E-mail*: parvanoj@bg.pims.org.

BACK TO TOP

e-mail: infosec@mbox.digsys.bg

Authors: **G.C. Fox, W. Furmanski, B. Natarajan, H. T. Ozdemir, Z. Odcikin Ozdemir, S. Pallickara, T. Pulikal**
Title: **Integrating Web, Desktop, Enterprise and Military Simulation Technologies to Enable World-Wide Scalable Televirtual (TVR) Environments**
Year of issuance: **1999**
Issue: **Information & Security. Volume 3, 1999**
Hard copy: **ISSN 1311-1493**

INTEGRATING WEB, DESKTOP, ENTERPRISE AND MILITARY SIMULATION TECHNOLOGIES TO ENABLE WORLD-WIDE SCALABLE TELEVIRTUAL ENVIRONMENTS

G.C. FOX, W. FURMANSKI, B. NATARAJAN, H. T. OZDEMIR, Z. ODCIKIN OZDEMIR, S. PALLICKARA and T. PULIKAL

Table Of Contents:

---

1. Introduction

Support for collaboration was one of the main driving forces for the original Web, created by CERN physicists to share data across high energy physics labs. Over the last decade, the Web technologies rapidly evolved and so did the collaboration capabilities of the whole framework. Natural initial focus was on asynchronous collaboratory models (shared databases) which were more recently augmented by the synchronous components (shared displays) and are now further evolving towards the ultimate televirtual (TVR) (shared worlds) environments. Some essential Web technology thresholds towards TVR included: VRML for 3D interactive front-ends and Java that enabled real-time synchronous connections.

First generation systems were represented by custom Java based collaboratory server technologies such as NCSA Habanero or JavaSoft JSDA. The recent onset of distributed object and/or component technologies opens new interesting avenues for the standards based collaboratory infrastructure. However, selecting a specific direction in the exploding field of distributed object and component technologies is not an easy task. CORBA offers one promising approach, especially from the large scale computing perspective but there are alternatives such as Java RMI or Microsoft DCOM. Finally, the World-Wide Web Consortium is developing a set of new standards such as XML, DOM, RDF and HTTP-NG which, when combined, can be viewed as yet another, new emergent distributed object model (sometimes referred as WOM [1]). It is likely that this model is more easily adopted by simple to medium complex distributed object/component applications.

Recent OMG/DARPA workshop on compositional software architectures [1] illustrated very well both the growing momentum and the multitude of options and the uncertainty of the overall direction in the field. A closer inspection of the distributed object/component standard candidates indicates that, while each of the approaches claims to offer the complete solution, each of them in fact excels only in specific selected aspects of the required master framework. Indeed, it seems that WOM is the easiest, DCOM the fastest, pure Java the most elegant and CORBA the most complete solution.

In our approach towards TVR, we adopt the integrative methodology, i.e. we setup a multiple-standards based framework in which the best assets of various approaches cumulate and cooperate rather than compete. We start the design from the middleware which offers a core or a 'bus' of modern 3-tier systems and we adopt Java as the most efficient implementation language for the complex control required by the multi-server middleware. We adopt CORBA as the base distributed object model at the Intranet level, and the (evolving) Web as the world-wide distributed (object) model. System scalability requires fuzzy, transparent boundaries between Intranet and Internet domains which therefore translates into the request of integrating the CORBA and Web technologies. We implement it by building a Java server which handles multiple network protocols and includes support both for HTTP and IIOP. On top of such Object Web software bus, we implement specific computational and collaboratory services.

2. Taxonomy of Collaboratory Frameworks

The most straightforward is an *asynchronous* collaboratory framework, typically offered by Web linked relational databases. Participants interact in such a shared space at their own pace, using some suitable connection identifiers to maintain the session identity across subsequent client-server connections. The atomicity, concurrency, security and other transactional capabilities of the database ensure consistency of the shared space. Our two projects, one in healthcare domain (Careweb) and another one in distance education (Language Connect University - LCU) used this type of collaboratory system based on CGI-extended web servers with tools for creating and editing shared documents in the database. In Careweb, we used a centralized Oracle database of student health records to bring various members of the school health community (school nurses, nurse practitioners, pediatricians, parents, etc.) together. LCU also used CGI-based approach for bringing a virtual classroom of students and instructors together over the web. It used a hypermedia database where students can take lessons, quizzes, exams, etc. and instructors can grade these submissions and respond back to the students in various different ways. Support was also there for a Web/Oracle based customized mailing facility wherein various members of the virtual university can interact with others in a group or on an individual basis.

The onset of Java and ActiveX brought full dynamic/remote programmability to the client/Web browser side and hence the support for the socket based active connections between clients and collaboratory servers. This enabled *real-time synchronous* collaboratories, typically implemented as a collection of Java applets, maintaining active connections to the common server-side shared/collaboratory object and updating their displays within some synchronous communication model such as chat, whiteboard, webcast etc. The current first generation of such systems offers various API solutions, ranging from simple generic collaboratory Channel model of JSDA by JavaSoft, to the full collaboratory applet (hablet) model of Habanero by NCSA. In our collaboratory experiments, we explored both JSDA and Habanero models from the perspective of their relevance for TVR collaboratory environments.

Televirtual (TVR) frameworks are based on the shared virtual world metaphor, popular in the networked virtual reality community. Hence, TVR can be viewed as a natural merger of asynchronous and synchronous collaboratory technologies discussed above. Participants get engaged in synchronous collaboration within their individual 'reality windows' whereas the rest of their worlds follow the asynchronous update and access model. Naively, one could implement TVR simply as a central server based synchronous collaboratory with clipping but such solution clearly does not scale towards truly large multi-user worlds. A natural scalable solution is given by a peer-to-peer or a multi-server model in which each participant joins the world with her/his own 'personal' server. Distributing an active multi-user virtual world over a set of servers raises in turn several non-trivial synchronization issues. Such parallel or distributed simulation environments were actively investigated during the last decade by the DoD. A set of promising solutions and standards is now available, including DIS, SPEEDES and most recently HLA/RTI. These systems offer sophisticated algorithms for managing logical simulation time and handling the distributed synchronization such as dead reckoning in DIS, Global Virtual Time or Breathing Time Warp in SPEEDES, and Data Distribution Management in HLA/RTI.

3. Towards Non-Trivial Collaboratory Applications

Simple Web based collaboratory applications are already being used by specific communities such as on-line gaming, chat rooms, or distance learning. Here, we are trying to address a more complex question of what comprises a useful collaboratory application for an enterprise or a research lab. Such institutions often use computers to address and solve some complex problems in their domain. Hence, for collaboration to succeed, it needs to be made part of the mainstream computational framework used by an organization, rather then a standalone or add-on capability, which is typical to the current generation systems.

Such integration of collaboration and computation can be most naturally addressed within the current models of distributed objects such as CORBA, Java RMI or DCOM. Indeed, a shared entity such as a CORBA object can act as a natural collaboratory channel, while at the same time it can be associated with some server side processing or it can interact with other computational CORBA objects. One example of such approach is given by the WebFlow environment under development at NPAC.[2,3]

WebFlow is a 3-tier distributed visual dataflow model. WebFlow front-end is given by a Java applet that offers a visual interactive tool for dataflow authoring. WebFlow middleware is given by a mesh of Java Web Servers that offer servlets based management of the distributed computation: Session Manager that interacts with the editor applet, Module Mamanger that instantiates computational modules represented as visual dataflow nodes, and Connection Manager that communicates with other WebFlow servers to form distributed computational meshes of WebFlow modules. WebFlow Module is a Java Object which implements webflow.Module interface with three methods: init(), run(), destroy(). WebFlow backend is currently left open for experimentation with various computational (e.g. HPF, RDBMS) and collaboratory (e.g. JSDA) paradigms. This allows us to put computational and collaboratory components of a complex computational environments into the common framework of visual dataflow authoring. Some natural collaboratory applications under development within this model include: collaboratory visual software engineering (collaboratory UML front-ends); simulation based design (collaboratory VRML authoring), visual HLA simulation tools (collaborative authoring of federations or federate objects for wargaming simulations).

4. Enabling Technologies

We follow the 3-tier architecture with the distributed object/componentware based middleware, visual front-ends and suitable simulation, computation or information/database objects in the back-ends.

In the front-end, the most promising collaboratory technologies include: Java3D, VRML, DirectX, XML. Is seems that Java3D and VRML communities will join their efforts, but it is not clear yet how these technologies are to be integrated with the Microsoft approach based on XML application languages on top of DirectX engines.

Virtual Reality Modeling Language (VRML) is a file format that defines the layout and content of a 3D world with links to more information. The Java 3D API is an application programming interface used for writing stand-alone three-dimensional graphics applications or Web-based 3D applets. DirectX is a group of technologies designed by Microsoft to make Windows-based computers an ideal platform for running and displaying applications rich in multimedia elements such as full-color graphics, video, 3D animation, and surround sound.

In the middleware, we are exploring CORBA, Java/RMI and DCOM and we are addressing their integration towards a uniform Object Web software bus. We are also augmenting this generic layer of distributed objects and components by specialized DoD technologies such as SPEEDES, DIS and HLA/RTI in the distributed simulation domain. These technologies are being discussed in the following sections in details.

In the back-end, we are analyzing simulation object technologies such as HLA/FOM or HLS/SOM, the generic CORBAservices and CORBAfacilities, and the transparent persistence technologies such as Java JDBC, CORBA PSS or Microsoft OLEDB which are still evolving but essential for building large and heterogeneous and yet scalable worlds.

JDBC API from JavaSoft, is a standard SQL database access interface for accessing heterogeneous databases from Java programs. It encapsulates the various DBMS vendor proprietary protocols and database operations and enables applications to use a single high level API for homogeneous data access. With the integration of Java and databases, many visually challenging, collaborative applications can be developed with ease.

One of OMG's Corba Object Services, Persistent State Service (PSS), addresses the issues of making persistent CORBA objects across machines, platforms and datastores. PSS provides the platform for storing and managing distributed business objects over heterogeneous datastores in a reliable and scalable manner for general and common shared use. PSS makes use of other CORBA services like Transaction Service and features like portable object adapter, objects-by-value etc. to give a location transparent, language and datastore-independent access to the objects.

OLEDB, which is the core of Microsoft's Universal Data Access strategy, defines a set of COM interfaces by which data providers, consumers and service components can interact with ease, for developing multitier enterprise applications. Applications or service components like query processor, cursor engine etc. can access the underlying diverse data in a unique way. The ActiveX Data Objects built on top of OLEDB gives a language and data provider - neutral, extensible and easy to use way for manipulating the data. Thus, applications can use the same interface to access various heterogeneous datastores like mail stores, project management tools, ODBC databases etc. While OLEDB addresses persistency of COM objects and PSS that of CORBA objects, we are trying to integrate these two technologies to develop a location, language, operating system and datastore-independent, transparent way of persistent object storage.

5. Technology Integration: A Case Study

Our early TVR experiments [4] were based or JSDA collaboratory server and VRML multi-user front-ends, linked to the JSDA channels via the EAI or the Java scripting. The associated computational experiments within the WebFlow model were using Java Web server middleware and Java applet authoring front-ends.

We are currently building the new version of an Object Web based collaboratory WebFlow environment which includes and integrates several enabling technologies listed above. Our current application domain that drives the system design and prototyping is given by the WebHLA based military modeling and simulation where we are developing the Object Web based RTI and the WebFlow based visual authoring tools for HLA simulations. The software bus of our system is given by JWORB - Java Web Object Request Broker that integrates Java, Web and CORBA middleware technologies.

JWORB is a multi-protocol extensible server written in Java. The base server has HTTP and IIOP protocol support. It can serve documents as an HTTP Web Server and it handles the IIOP connections as an Object Request Broker. As an HTTP server, JWORB supports Servlet and CGI mechanism. Any servlet developed with Java Servlet API can run with JWORB.

Since JWORB design is Object Oriented, it is very easy to add other protocols. As JWORB starts up, it looks at configuration files to figure out which protocols it is capable of handling and it loads the necessary protocol classes for each protocol. If we want to add a new protocol, we need to implement a few abstract classes defined for the protocol object and to register this protocol implementation in the configuration file. This mechanism allowed us to define HTTP and IIOP protocols in the current prototype and we are investigating to include the DCOM Protocol or CORBA-to-DCOM bridge to be able to communicate with the DCOM objects.

On top of JWORB, we are developing dedicated services such as CORBA collaboratory.

5.1. Collaboration based on CORBA

We include below the two most significant IDL definitions in our proposed CORBA Collaboration Service. The IDL definitions signify the operations a Client could invoke on a remote instance of these objects. Invocation of any of these aforementioned operations should be preceded by a successful reception of a remote handle to these objects.

```
interface Coordinator {
   boolean setMaxClients(in long arg0);
   long getMaxClients();
   long numberOfMembers();
   typedef sequence string sequence_of_string;
   MultiCoordinator::Coordinator::sequence_of_string
                                  getClientNames();
   boolean isEmpty();
   long register(in long arg0, in string arg1,
                    in Client::ClientControl arg2);
   boolean deregister(in long arg0);
   boolean broadcast(in string arg0);
   boolean whisper(in string arg0, in long arg1);
};

interface PartyScheduler {
   boolean createParty(in string arg0);
   long getPartyID(in string arg0);
   MultiCoordinator::Coordinator
                  getPartyHandle(in long arg0);

};
```

The PartyScheduler is the one which schedules the appropriate instance of the Coordinator Object to coordinate Clients logged onto a specific session (Party) comprising of possible different applications. In the event that there is a Distributed Directory service and the Active Object server is in place, the Client is ready to invoke the IDL-defined operations.

1. It starts with the createParty(String partyName) function which would return a true in the event that a new Coordinator Object has been instantiated or a false to

signify the prior existence of the desired party.

2. The Client gets a handle to the Coordinator Object by invoking long getPartyID(in string arg0); MultiCoordinator::Coordinator getPartyHandle(in long arg0); in succession.
3. Once Steps I and II are over and done with, the Client is now in a Distributed Collaboration mode, and can invoke operations specified in IDL definitions for the Coordinator.

5.2. RMI-Based Collaboratory

The abstractions provided by the CORBA Collaboratory are maintained in the RMI version, except that instead of IDL definitions as the starting point Java Interfaces perform the same function.

```
import java.rmi.*;

public interface PartyScheduler extends Remote {
  boolean createParty(String PartyName)
                throws java.rmi.RemoteException;;
  int getPartyID(String PartyName)
                throws java.rmi.RemoteException;;
  MultiCoordinator.Coordinator getPartyHandle(int partyID)
                throws java.rmi.RemoteException;;
}

package MultiCoordinator;
import java.rmi.*;
public interface Coordinator extends Remote {
  boolean setMaxClients(int _maxClients)
           throws java.rmi.RemoteException;
  int getMaxClients()throws java.rmi.RemoteException;
  int numberOfMembers()
           throws java.rmi.RemoteException;
  String[] getClientNames()
           throws java.rmi.RemoteException;
  boolean isEmpty() throws java.rmi.RemoteException;
  int register( int clientHashCode, String ClientName,
            String clientObjRef)
            throws java.rmi.RemoteException;
  boolean deregister(int clientID)
            throws java.rmi.RemoteException ;
  boolean broadcast(String Message)
            throws java.rmi.RemoteException;
  boolean whisper(String Message, int clientID)
            throws java.rmi.RemoteException;
}
```

The RMI based Collaboration is about 30-40 % faster than the IIOP solution, however the advantage RMI holds is blunted by the fact that its a pure Java Solution. RMI-Collab is platform independent, albeit expressed through Java. Nevertheless, CORBA is a platform independent and language independent solution. With CORBA one could have Java Helper classes accessing a C++ implementation of the Party Scheduler. The choice is clear in case of pure Java solutions write to RMI else write to CORBA.

5.3. Logical Time and Data Distribution Management

Communication locality is the most important concept which allows us to build large scale scalable collaborative environments. This essential feature is enabled via event filtering in terms of JSDA channels, CORBA Event Service and RTI routing spaces.

CORBA's Event Service tries to address the Subscribe/Publish pattern for information exchange between objects. By relying on this technology, it is possible that each participant keeps its Event Channel as its broadcasting channel in his/her machine and publishes its Channel address in the public directory so that all interested parties can get the address of this Event Channel and subscribe this channel.

In HLA/RTI,[6] DDM [5] allows simulations to define a routing spaces so that the communication layer delivers the interaction and attribute updates to the appropriate simulation unlike broadcasting to everybody [7] and consuming computation time and network bandwidth [8] defined by Distributed Interactive Simulation (DIS).[9]

Time Management [10] service of RTI is being developed to provide appropriate time stamps for messages while allowing a simulation to pick the appropriate message order (FIFO, Priority Based, Total Order, and Causal Order). Handling messages in the correct order is a challenging problem since participants are distributed and network introduces latency. Because of these delays participants can receive messages which were sent in the past but arrived late.[11] The solution for this problem is to provide a synchronization mechanism between participants.

One of the approaches is to synchronize everybody conservatively. Conservative approach spends a lot of time for synchronization and wastes network bandwidth and computation resources. The second alternative known as Optimistic approach allows participants to proceed into the future (for a small time window); if they receive an event from past then they rollback. The last synchronized time is defined as a Global Virtual Time.[12] The determination of Global Virtual Time (GVT) consumes network traffic and time. Therefore, we wish to calculate GVT at the lowest possible frequency.[13] To find the GVT, the minimum time stamped event (message) has to be determined while taking care of the events (messages) in transit. SPEEDES [14] handles this problem by counting the sent and received events globally in the system. Whenever it finds out they are equal, then it starts the process related to determination of GVT.

5.4. Visual Authoring Tools

Front-end of our WebHLA prototype will include visual authoring tools for the HLA simulation objects. HLA Simulation Tools are being designed to provide automated support for development of HLA Object Models (OM), generation of RTI federation execution data and exchanging OMs with the Object Model Library. Currently, available tools include Aegis Object Model Development Tool (OMDT),OSim from OriginalSim Inc

Currently, our efforts in this area include development of a Simulation tool that conforms to the OMDT look-and-feel and it is included as a module/component in the WebFlow framework. Also, experiments are being performed to explore the feasibility of exploiting useful components like Microsoft Excel'97 spreadsheet through its COM interfaces for the purpose of tree-like graphical representation of the simulation object model.

5.5. DirectX meets HLA

The DirectX is a common infrastructure from Microsoft Inc., in the implementation of high performance real-time applications such as computer games and multimedia applications. The component of DirectX that supports multiplayer, networked applications is called DirectPlay. The functions provided by DirectPlay are similar to those provided by the HLA RTI with a few significant differences. DirectPlay provides some features specific to gaming. However, it does not provide any time synchronization features; it is built to support a DIS-like, loose causality model. DirectPlay also differs from the RTI in that it does not provide any support for determining data routing. In DirectPlay, there is no concept of a common object or data definition.

DirectPlay is possibly the more suitable technology for multi-user PC games. However combining DirectPlay with concepts of DIS and HLA might create a hybrid technology that could be useful for both military simulations and the entertainment industry.

We are currently designing the runtime display support using VRML and DirectX technologies, linked to the Java middleware via suitable interfaces (EAI, Java scripting, COM/CORBA bridge). We are also analyzing a set of advanced DoD simulation kernels including RTI, SPEEDES and ModSAF and we intend to experiment with their underlying time management algorithms within our JWORB framework via suitable CORBA interfaces. Thus, the JWORB middleware acts as a universal software bus linking desktop/commodity visualization front-end with the DoD M&S backend.

6. Summary

We believe that the current suite of Web/Commodity technologies provides us with a critical mass sufficient to build the next generation world-wide scalable televirtual environments. We outlined here several enabling technologies that need to be integrated and we summarized the current status of our prototyping experiments. At the moment, our Java Web Server based WebFlow prototype is operational and we are also testing the early prototype of our new JWORB middleware. We are also experimenting with and evaluating the front-end display (VRML/Java3D/DirectX) and the back-end persistent store (JDBC/PSS/OLEDB) technologies as they emerge and evolve. In parallel with this core technology R&D, we are also exploring a suite of advanced military simulation environments such as SPEEDES, ModSAF, HLA/RTI. Early integration demos for multi-user WebHLA authoring and runtime are already available.

References

1. Craig Thompson, *OMG/DARPA Workshop on Compositional Software Architectures* (Monterey, CA, January 6-8, 1998), Available at http://www.objs.com/workshops/ws9801
2. D. Bhatia, V. Burzevski, M. Camuseva, G. Fox, W. Furmanski and G. Premchandran, "WebFlow - a visual programming paradigm for Web/Java based coarse grain distributed computing," in the special issue of "*Concurrency: Practice and Experience*" on Java for Scientific Computing (February 1997).
3. G. Fox, W. Furmanski and T. Haupt, "ASCI WebFlow: High-Level Programming Environment and Visual Authoring Toolkit for High Performance Distributed Computing," Available at http://www.npac.syr.edu/projects/asci-webflow
4. D. Dias, G. Fox, W. Furmanski, V. Mehra, B. Natarajan, H. T. Ozdemir, S. Pallickara and Z. Ozdemir, "Exploring JSDA, CORBA and HLA based MuTech's for Scalable Televirtual (TVR) Environments," presented at *VRML98* (Monterey CA, February 1998).
5. Danny Cohen and Andreas Kemkes, "Using DDM - an Application Perspective," *1997 Spring Simulation Interoperability Workshop (SIW)* (1997), 97S-SIW-014
6. High Level Architecture (HLA) by the Defence Modeling and Simulation Office (DMSO). See also http://www.dmso.mil/hla
7. Katherine L. Morse, *Interest Management in Large-Scale Distributed Simulations*, Information and Computer Science Technical Report, ICS-TR-96-27 (UC Irvine, 1996).
8. Duncan C. Miller, "The DOD High Level Architecture and The Next Generation of DIS," *14th DIS Workshop* (March 1996), 96-14-115.
9. *IEEE Standard for Distributed Interactive Simulation - Application Protocols*, IEEE 1278.1-1995
10. HLA Time Management Design Document (August 16, 1996). Available at http://www.dmso.mil/hla/tech/ifspec
11. L. Lamport, "Time, Clocks, and the Ordering of Events in a Distributed System," *Communications of the ACM* 21, 3 (July 1978), 558-565.
12. D.R. Jefferson, "Virtual Time," *ACM Transactions on Programming Languages and Systems* 7, 3 (July 1985), 404-425.
13. J. Steinman, C.A. Lee, L.F. Wilson and D.M. Nicol, "Global Virtual Time and Distributed Synchronization," In *Proceedings of the 1995 Parallel and Distributed Simulation Conference* (PADS'95), 139-148. Available at http://www.acm.org/pubs/contents/proceedings/simulation
14. J. Steinman, SPEEDES: Synchronous Parallel Environment for Emulation and Discrete-Event Simulation, In *Proceedings of the SCS Western Multiconference on Advances in Parallel and Distributed Simulation* (PADS91), vol. 23, 1 (1991), 95-103. Available at http://www.acm.org/pubs/contents/proceedings/simulation

---

G.C. FOX, W. FURMANSKI, B. NATARAJAN, H.T. OZDEMIR, Z. ODCIKIN OZDEMIR, S. PALLICKARA and T. PULIKAL are currently with the Northeast Parallel Architectures Center (NPAC) at Syracuse University. E-mail: {gcf, furm, balaji, timucin, zeynep, shrideep, tapulika}@npac.syr.edu

BACK TO TOP

Information & Security. An International Journal
e-mail: infosec@mbox.digsys.bg

Authors: **Valentine Penev, Tatiana Atanasova, Ivanka Valova**
Title: **Distributed Simulation and Modeling Environment**
Year of issuance: **1999**
Issue: **Information & Security. Volume 3, 1999**
Hard copy: **ISSN 1311-1493**

DISTRIBUTED SIMULATION AND MODELING ENVIRONMENT

Valentine PENEV, Tatiana ATANASOVA and Ivanka VALOVA

Table Of Contents:

---

1. Introduction

Distributed simulation has become a widely popular and useful tool for various applications, including military simulation and training, for example, ModSAF.

The goal of Distributed Simulation and Modeling (DSM) is to create a virtual world incorporating various simulators distributed over the network. It needs to manage different kinds of time infrastructures and communications strategies, based on real-time information processing.[1,2] A rich variety of simulators for different purposes have already been developed, but the integration of these simulators into a common framework needs further investigation. The work consists of simulation, seamless integration, and distributed virtual environments. The virtual environment includes its two-dimensional representation, three-dimensional visualization, physical simulation, behavior, and networking.

In this paper a distributed environment which shares the computational resources is proposed. Distributed simulation systems use models, visual representation and calculation of object movements. The network distribution allows remote access to heterogeneous data sources and functionally divided operation of the system. The status of the environment and its parts are updated and sent over the network so that all participants are fully aware of the environmental situation. Different optimization algorithms are used for tasks of simulation, computation, and visualization.

2. Distributed System for Modeling and Simulation

Simulations are very complex systems with many interactions between the involved variables.[2] A lot of computations is needed to perform the various simulation tasks. Figure 1 presents the proposed scheme for modeling and simulation system. Different parts of the synthetic environment with their corresponding algorithms are distributed among several computational resources connected by a network.

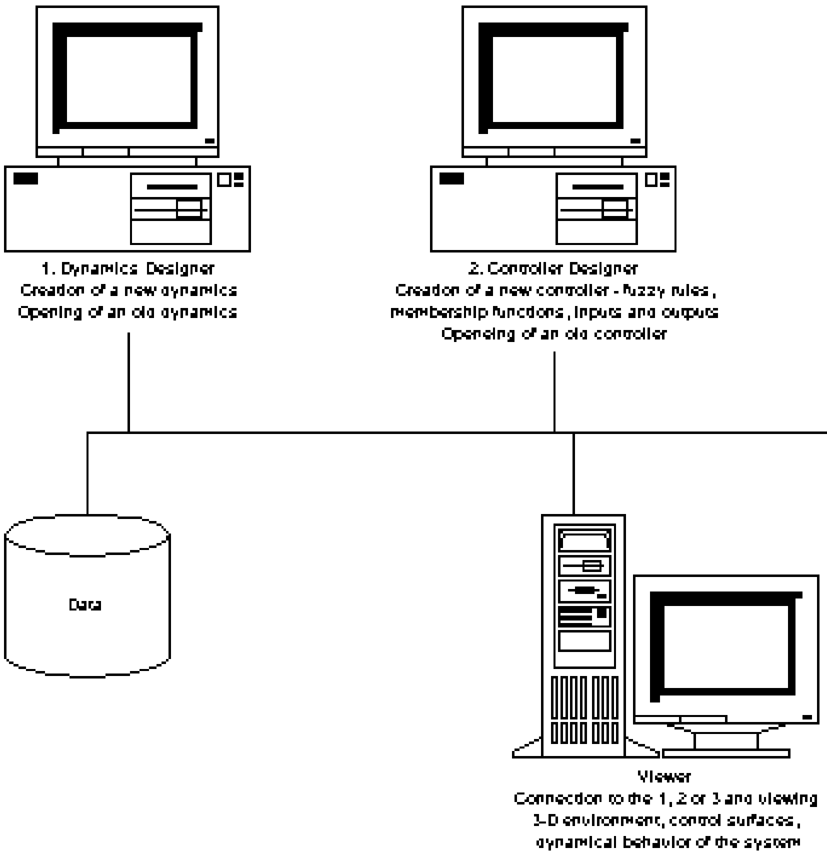


Figure 1. DIS architecture

The blocks in this scheme have the following functions:

Dynamics Designer (DD) of the simulated objects:

- Creates a new dynamics set of modeling objects
- Reads and shows old dynamics set
- Defines the dynamics set inputs for test
- Saves a dynamics set
- Displays dynamics set and dynamical response to the inputs
- Saves dynamical response

Controller Designer (DC):

- Creates a new controller
- Reads and shows old controller
- Defines controller input for test

- Saves controller
- Displays controller and controller output behavior to the inputs
- Saves controller output behavior

The primary function of this block is to create appropriate controller for the defined simulated object.

Decision Support Unit (DSU):

Performs selection of the best regulator, dynamics, open loop system, closed loop system on the base of defined criteria. DSU consists of several integrated parts with approximate reasoning capabilities. The system provides fuzzy reasoning combining fuzzy sets, hedges and fuzzy if-then rules into non-procedural collections – policies.

Viewer defines connections between Distributed Knowledge Data Base, DD, DC and DSU:

- Creates new loop (dynamics, controller). The loop consists of the following items – Dynamics, Controller and connections between them.
- Reads and shows old loop
- Defines input to the loop
- Saves loop
- Displays loop dynamical response to the inputs
- 3D – viewer of behavior of object and control surfaces
- Saves dynamical response of loop

Viewer utilizes the information about the simulated object, its dynamics, behavior and environmental conditions. Solutions for the visualization include interactive real-time 3D displays of virtual worlds in a range of graphics tools and interfaces including Direct3D and others.

Distributed Knowledge Data Base (DKDB) is represented by series of data elements, rules, documents and software modules. DKDB keeps the following kinds of data:

- Dynamical set - equations and numerical coefficients.
- Inputs to Dynamical Set - step, linear, sinusoidal, joystick.
- Controller - The controller has the following characteristics: inputs and outputs, fuzzy input and output variables, membership functions, fuzzy rules and defuzzification procedure.
- Dynamical responses - time sequences of dynamics outputs and state variables.
- Environmental data. An essential step in environmental modeling is the collection or creation of data to represent the specific type of environment required. This includes information about the terrain, atmosphere, ocean surface, floor, vegetation, imagery, etc. The list is extensive and the storage requirements are large. The terrain database is distributed over the network so that any simulator entering the virtual world could get background information from the network.

Data units are sent over the network by addressing them to a particular computer and preprocessing the data to satisfy some specific types recognized by the receiving program. The calculation process is implemented using modular applets (ActiveX) which automatically gather the corresponding data over the network. The simulation also receives or sends data produced from user inputs.

3. Description of the database

Entity in the DKDB describes any physical or virtual object defined by a set of properties such as its geometrical dimensions, geographical position, dynamical response, behavior, modeling parameters, etc. It has attributes and functions (variables and methods). The attributes are as following:

- Entity ID: a unique reference identifying each object.
- data field: it varies for different categories of entities. The field contains two sorts of information: parametric data and simulation data. Parametric data could be downloaded dynamically from a remote data base at the initialization of this kind of entity. The simulation data would be initialized at the creation of the entity through its parent. It consists of the following information:
    - entity type: a universal type identifier
    - status
    - time step and time synchronization
    - representation (from parametric data): properties used for entity modeling and simulating
- link to the related entity
- universal locator: a unique address

The entities in the DKDB are dynamical models of simulated objects, controllers, a set of views, predicted trajectory, documents, graphics, methods, software modules.

3.1 Functions of the entities

The methods attached to an entity differ depending on the entity. Some common functions that each entity implements are:

- Initialization of the parametric and simulation data through the remote data bases
- Update data field
- Update entities
- Get data
- Create entity
- Remove entity
- Change computational resource
- Specific - this category of function depends directly on the type of entity

This structure enables an entity to have function. It is useful for the surface modeling, because in this case the entity uses its own modeling algorithm, without depending on whether it would be grass, sand, or soil. For military applications, this could enable a terrain to have mines, hydrology models, vegetation, ground characteristics, instead of just a color or a texture map.

The description of data is based on HLA - the High Level Architecture. HLA is the standard technical framework for U.S. Department of Defense simulations. The HLA specifies the interface that simulations must use to communicate. This interface is implemented in the Run-Time Infrastructure (RTI).[3]

The developed services take advantage of network processing to improve distributed simulation accuracy, scalability, and performance. Beside this the computational and data caching, transport protocols, and optimized algorithms for locating and activating active code within a network are considered.

4. Integration of information applications in the DSM System

The interchange of information applications needs to be considered at four levels: data level; application program interface level; method level; and user interface level.

Data-level approach is concerned to the information extracting from one database, processing that information as needed, and updating it in another database. *Method-level approach* is the sharing of the logic that exists within the system. By sharing the logic or methods contained inside or outside any given application it is possible to share both data and processes among many applications, and therefore integrate the applications. *Application program interface-level* supports the interfaces to access both simulation processes and simple information (data). Using these interfaces, many applications work together, sharing logic and information to access processes and data, extract the information, place it in a format understandable by the target application, and transmit the information. Message brokers are one of the solutions, they are able to move messages among different parts of the system to reformat the messages, so they are understood by the target application. At *User interface-level* information is integrated on the basis of user interfaces as a common point of integration.

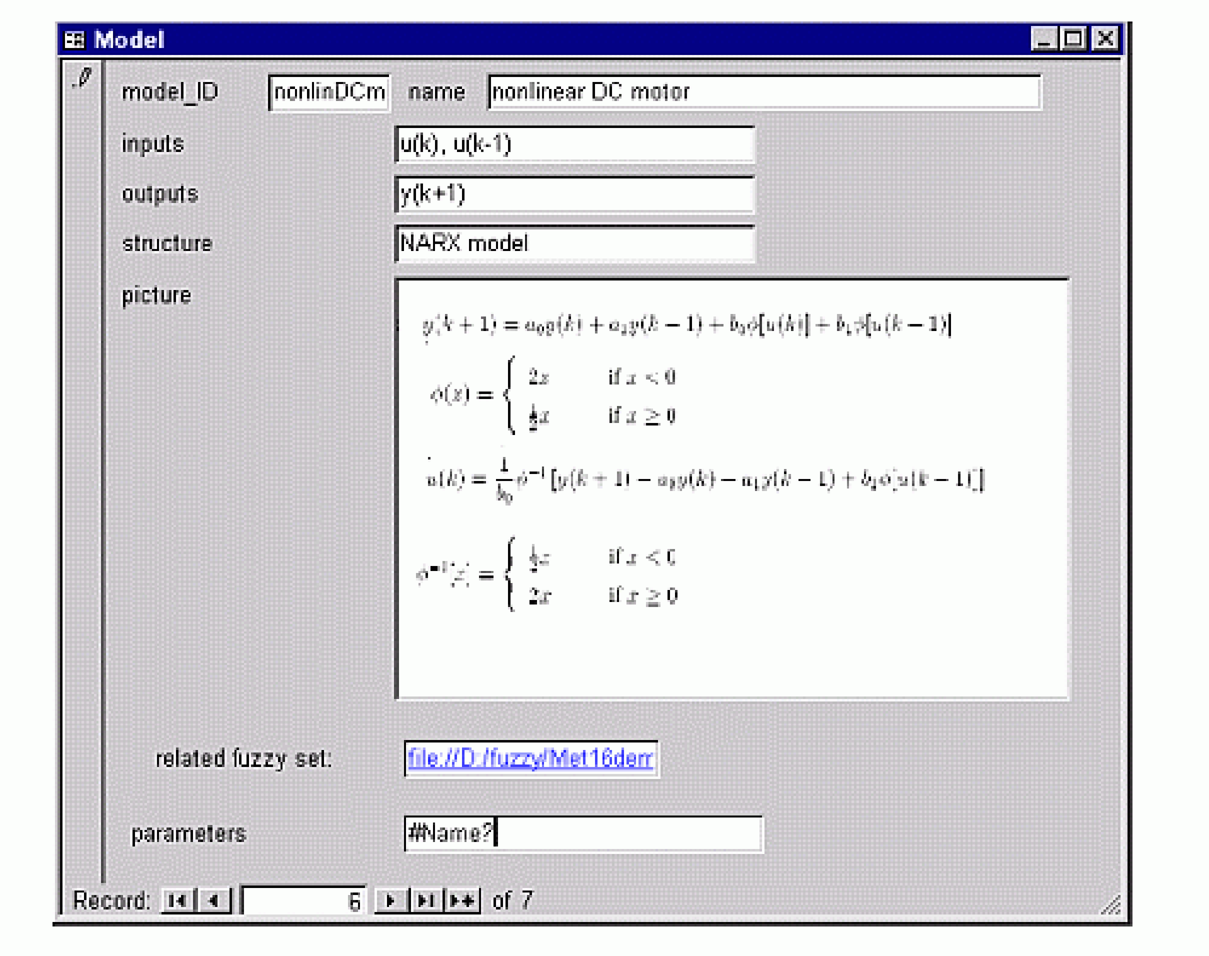


Figure 2. Preview of Dynamic Designer

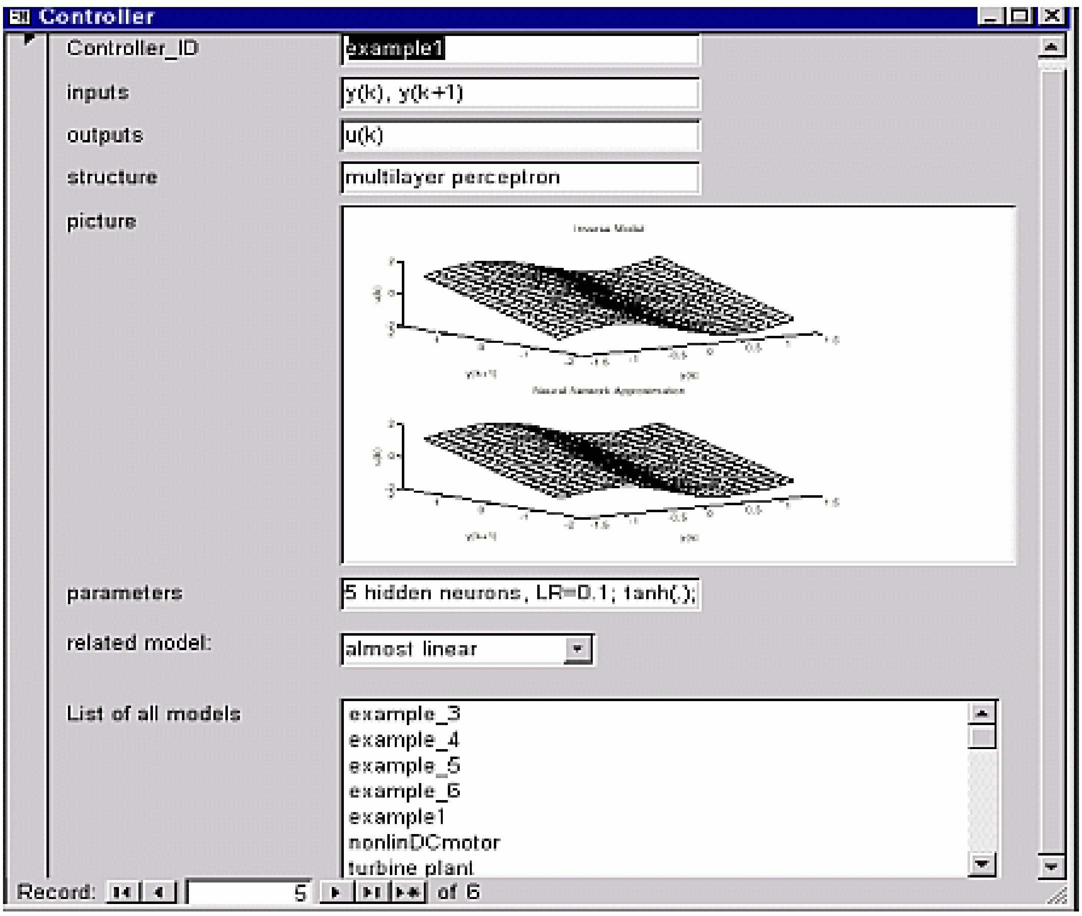


Figure 3. Preview of Controller Designer

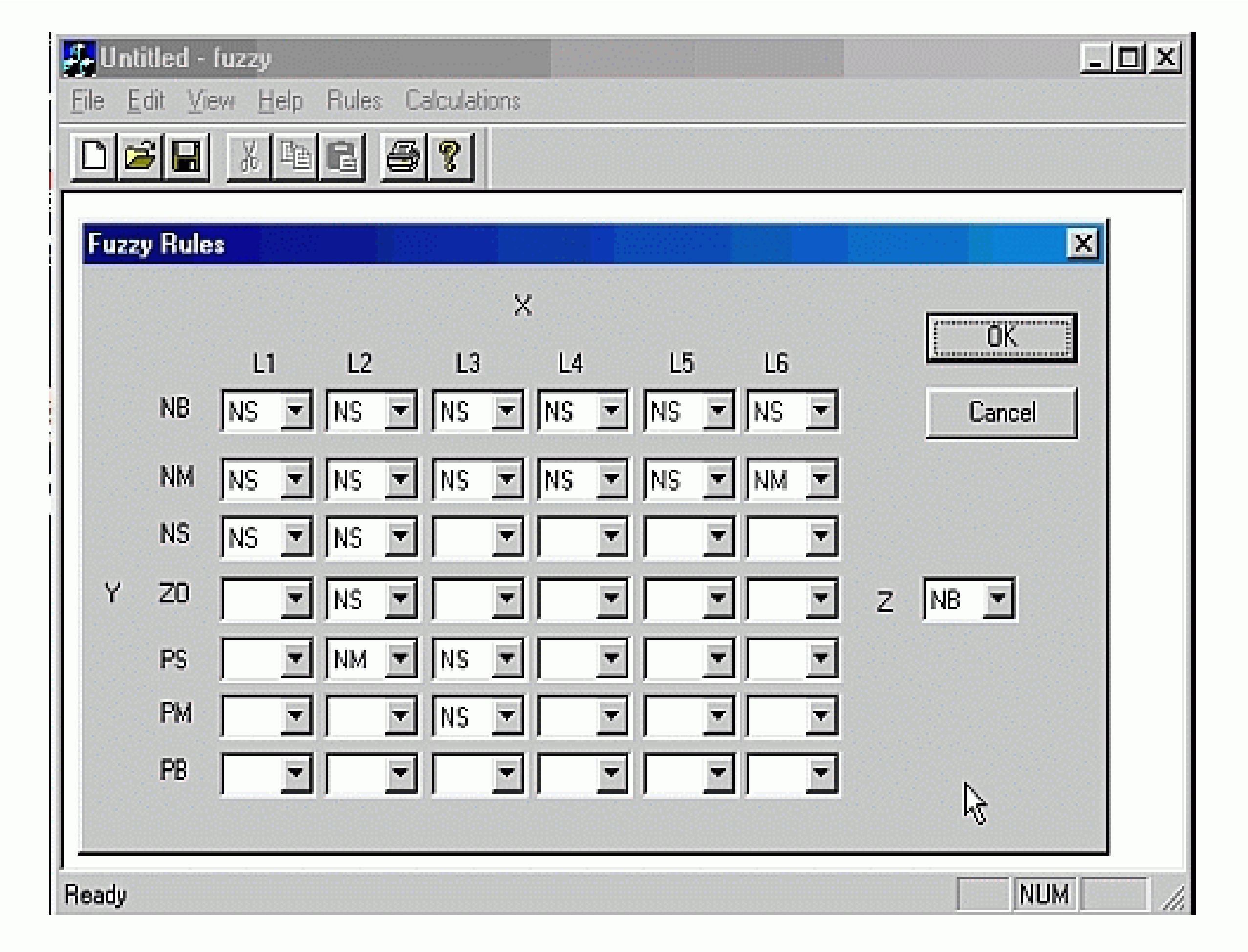


Figure 4. Preview of Fuzzy Rule for Controller Designer

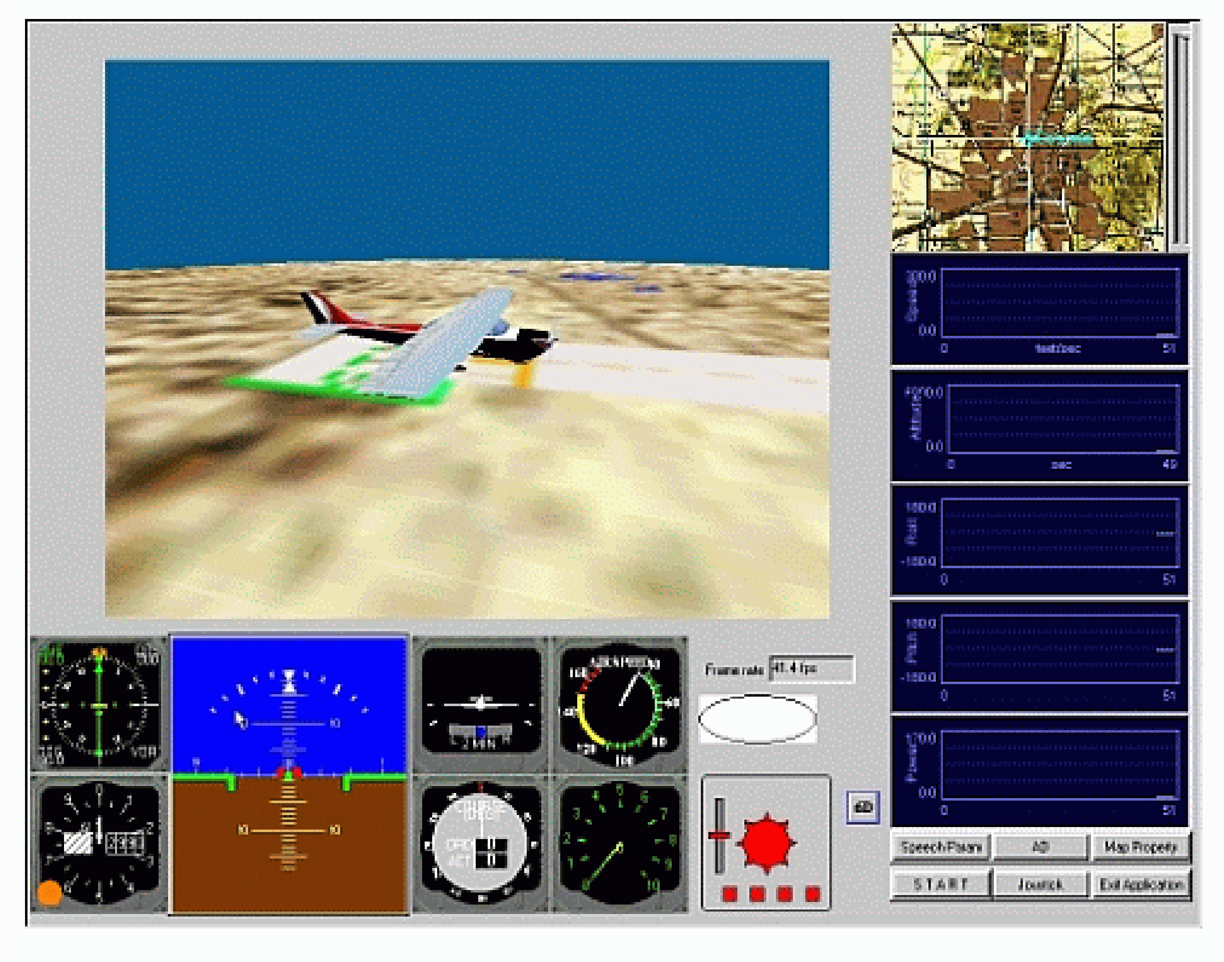


Figure 5. Preview of Viewer

4.1 Enabling Technologies

The integration of information is necessary for creating environments that realize collaboration with distributed computing and modern modeling and simulation technologies. It relies on the 3-tier architecture with the distributed object middleware, visual front-ends and suitable simulation, computation or information/database objects in the back-ends. In the front-end, the most promising collaboratory technologies include: Java3D, VRML, DirectX, XML.

The DirectX is a common infrastructure from Microsoft Inc. in the implementation of high performance real-time applications of modeling and simulations. The component of DirectX that supports multiplayer networked applications is called DirectPlay. The functions provided by DirectPlay are similar to those provided by the HLA RTI with a few significant differences. DirectPlay provides some features specific to gaming. However, it does not provide any time synchronization features; it is built to support a DIS-like, loose causality model. DirectPlay also differs from the RTI in that it does not provide any support for determining data routing. In DirectPlay, there is no concept of a common object or data definition.

DirectPlay is possibly the more suitable technology for multiuser PC games. However, combining DirectPlay with concepts of DIS and HLA might create a hybrid technology that could be useful for both military simulations and the entertainment industry.

Conclusion

We have presented a distributed Modeling and Simulation environment. The scalability limitations dictated by existing networking technologies continue to obstruct the achievement of high-performance, large-scale, distributed simulation. The objective of our future work is to exploit the novel capabilities of active networks to overcome these limitations through the development of advanced protocols and services specifically for distributed modeling and simulation. The future key research areas include using active network tools to provide dynamic management, computational and data caching in distributed data sources, and to interchange data from heterogeneous sources.

References

1. Roger Smith, *Military Simulation Techniques & Technology*, Detailed Course Outline. Available at http://www.simulation.com. An abridged version is published in the current issue of I&S.
2. Katherine L. Morse, *Interest Management in Large-Scale Distributed Simulations*, UC Irvine, Information and Computer Science Technical Report, ICS-TR-96-27.
3. IEEE Standard for Distributed Interactive Simulation - Application Protocols, IEEE 1278.1-1995.

VALENTINE PENEV Born 1960. M.Sc. (1986, Automatics and Telemechanics) from the Technical University of Tula, Russia. Ph.D. (1992, Fuzzy sets in digital control systems) from the Robotics and Control Systems Research Center at the Bulgarian Academy of Sciences. Currently Dr. Penev is Deputy Director for R&D of the Institute of Control and Systems Research at the Bulgarian Academy of Sciences. Senior Research Fellow with over forty publications in the field of intelligent technology engineering - fuzzy control, fuzzy reasoning, signal processing, and neuro-like structures. Software realization of a flight simulator and the simulator "Fagot" for ATGMs under Windows95, Windows 3.1x and DOS for PC. Address for correspondence: Institute of Control and Systems Research, Bulgarian Academy of Sciences, Acad.G.Bonchev Str., Bl.2, Sofia 1113, Bulgaria. E-mail: valpenev@iusi.bas.bg

TATIANA ATANASOVA received her Diploma in Automatics from the Power Institute in Moscow. Currently she is research associate at the Institute of Control and System Research. E-mail: tatiana@iusi.bas.bg.

IVANKA VALOVA is a researcher at the Institute of Control and System Research. E-mail: vania@iusi.bas.bg.

BACK TO TOP

Author: **Roger D. Smith**
Title: **Military Simulation: Techniques & Technology**
Year of issuance: **1999**
Issue: **Information & Security. Volume 3, 1999**
Hard copy: **ISSN 1311-1493**

MILITARY SIMULATION

Techniques & Technology

Roger D. SMITH

Table Of Contents:

"Military Simulation Techniques & Technology" is a 3-day training course designed to teach engineers and project leaders the essential techniques required to design, build, and operate a military simulation system.

The outline of the course is:

- Introduction to Simulation
- History of Simulation
- System Architecture & Design
- Interoperability
- Event Management
- Time Management

- Principles of Simulation
- Physical Modeling
- Environmental Modeling
- Behavioral Modeling
- Multi-Resolution Modeling
- VV&A

The topics in this course have evolved over several years of experimentation with different materials. These appear to be universally applicable to all simulation systems and most simulation projects. Each topic has some relevance to both the engineer and the manager.

<u>Course Principles:</u>

- Educate
  - Teach the most valuable and current simulation information available today
- Communicate
  - Present information in a form that can be understood
  - Provide materials that are useful in work environment
- Entertain
  - Maintain interest, enthusiasm, and participation
- Inspire
  - Trigger new ideas in the minds of students
  - Serve as the catalyst for innovation

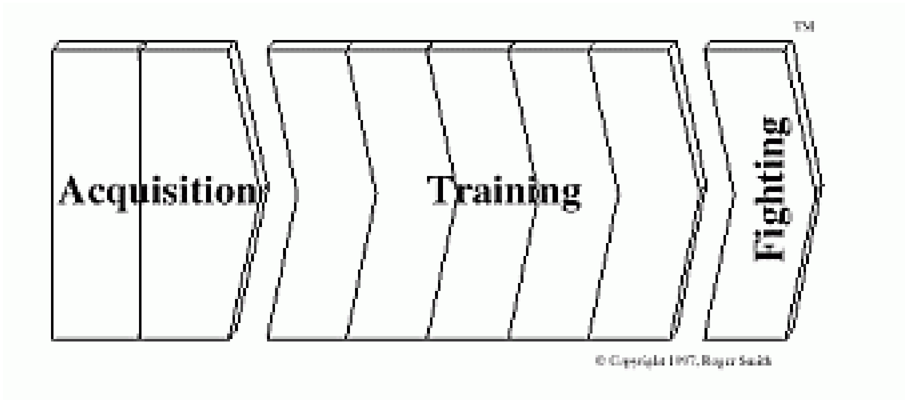
Fighting and Training

"Train the way we Fight" is not correct.

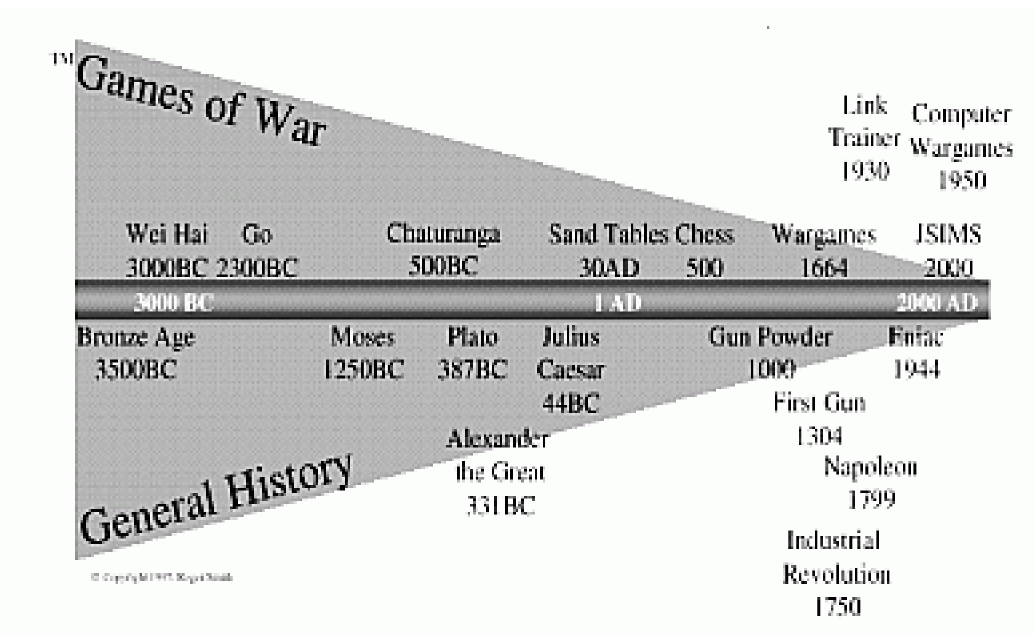
"Fight the way we Train" is more accurate.

Our society and government spends a great deal of time and money in the acquisition and training phases, long before entering into the fighting phase.

The experiences of each soldier during years of training have a direct impact on his or her performance when the fighting actually starts. If we provide inadequate or inaccurate training we are not preparing soldiers for real combat. This could result in the loss of their lives and, potentially, reduced influence of the United States in world affairs.



Games of War: A 5000 Year History



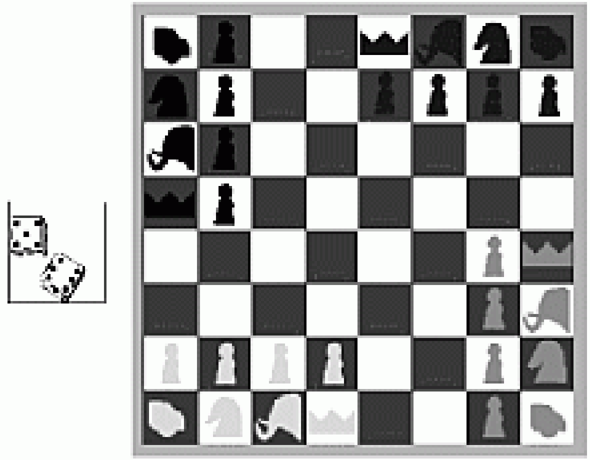
The history of military training goes back 5000 years to the Bronze Age. One may speculate that military training has existed as long as combat, armed conflict, and tribal feuds. The first references we can find to games of war are to Wei Hai around 3000 BC.

This timeline illustrates how long the history of this field really is. War gaming had its start long before the time of Moses.

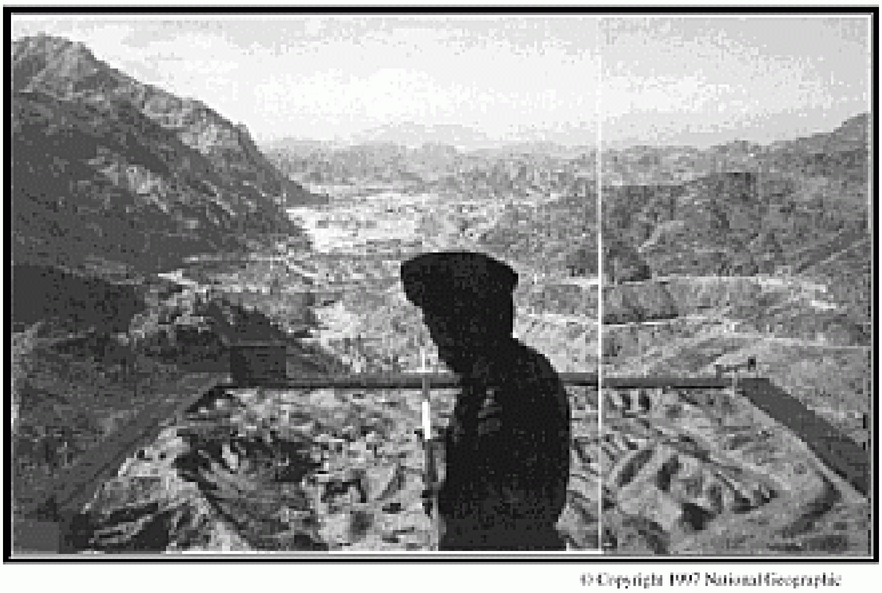
Chaturanga

The game of Chaturanga originates in India around 500 BC and is found in both 2-player and 4-player versions. The name literally means "four limbs", referring to the four branches of the military represented by the pieces. It makes use of pieces that roughly equate to those familiar to us in Chess. The pieces represent the Infantry, Rajah, Elephant, Cavalry, and Chariot (or Boatman). This game is the direct ancestor of Chess.

Originally the piece to move was selected by rolling dice. But Hindu law prohibited gambling and the dice were eliminated to avoid violating this law.



Pakistani Sandtable



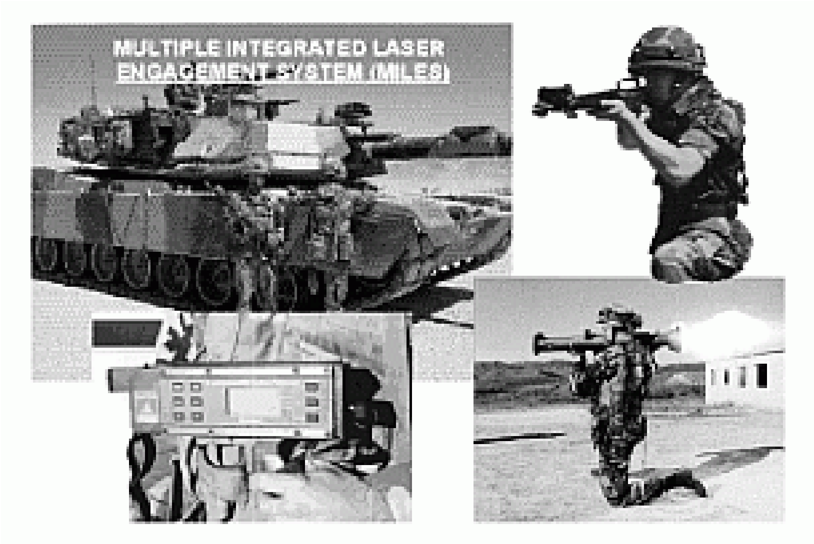
This sandtable overlooks the actual battlefield for the Khyber Pass, Pakistan.

From a safe vantage point commanders may experiment in miniature before committing their decisions to actual troops. Once commands are issued, the sandtable can be used to track the execution of these against the planned/expected outcomes.

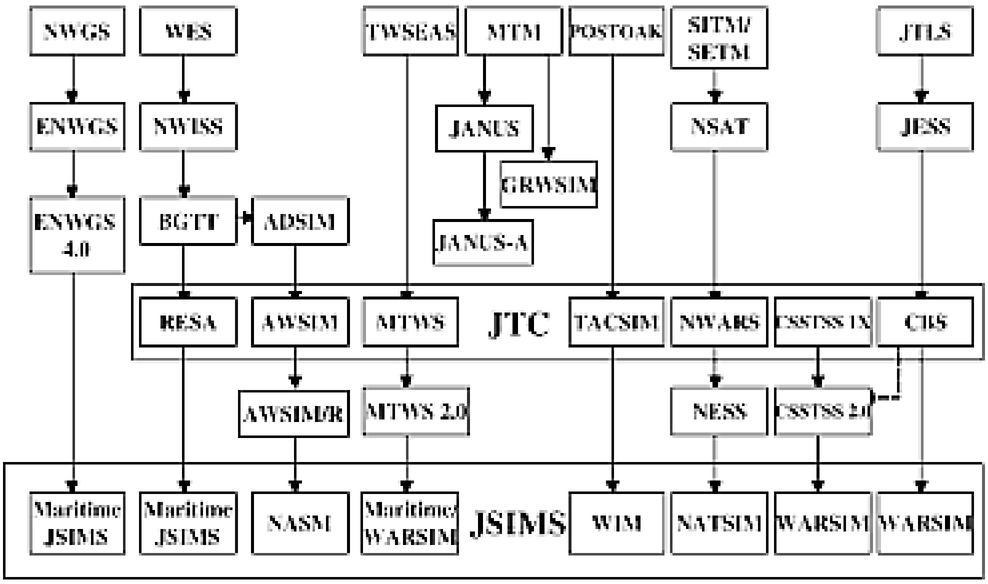
NTC Laser Warfare

The center of NTC force-on-force training is the Multiple Integrated Laser Engagement System (MILES). These are laser devices and sensors attached to each vehicle and soldier (e.g. 8 detectors on a soldier’s vest and 4 on his helmet). Laser beams are triggered when any weapon is fired. Microphones detect the sound of an exploding blank round and trigger the firing of a laser. This forces soldiers to load their weapon and limits them to actual magazine capacities.

If this laser beam hits the sensor on another vehicle or human a hit is scored. Killing one of these results in that weapon shutting down so it can no longer fire.



Warfare Model Evolution

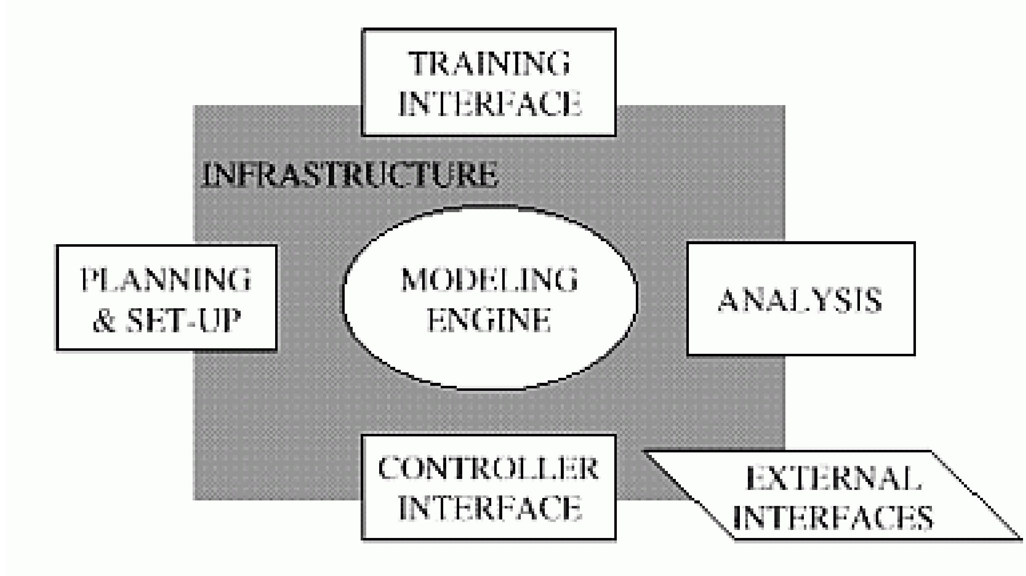


This is the model genealogy beginning in the early 1970’s and leading up to the current development of the members of the Joint Simulation System (JSIMS). The Joint Training Confederation (JTC) was an interoperability program that joined together several models that had originally been designed to operate independently. JSIMS is attempting to design the entire family to operate together from the beginning.

Joining models after they are created has proven to provide only a very limited degree of interoperability. Each model has a specific representation of the world that allows it to share/export information in very limited ways.

However, the JTC program proved that interoperability at this level is feasible.

Simulation Architecture



Following the establishment of six major components of a military training simulation, there evolved an infrastructure to tie all of these together. This infrastructure functionality had to be previously embedded in each of the components, usually in different forms with differing capabilities.

1) Modeling Engine - The representation of the objects, behaviors, and environments of interest in the training or analysis.

2) Infrastructure - Commonality of interfaces with the hardware, operating system, time management, event synchronization, network distribution, etc.

3) Planning & Set-Up - The process and tools used to prepare data, software, hardware, facilities, documents, and plans for an exercise.

4) Controller Interface - Interface that supports the organization, starting, stopping, and efficient progression of an exercise.

5) Training Interface - Interface that provides the training audience's experience of the simulation. This may be special displays and devices, C4I connections, etc.

6) Analysis - The process and tools used to organize and study the results of a simulation and the events that have occurred.

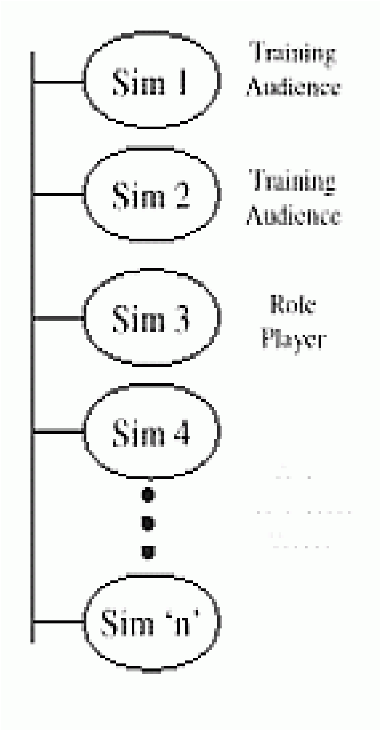
7) External Interfaces - Interfaces that allow one simulation to operate or exchange information with another.

Distributed Interactive Simulation

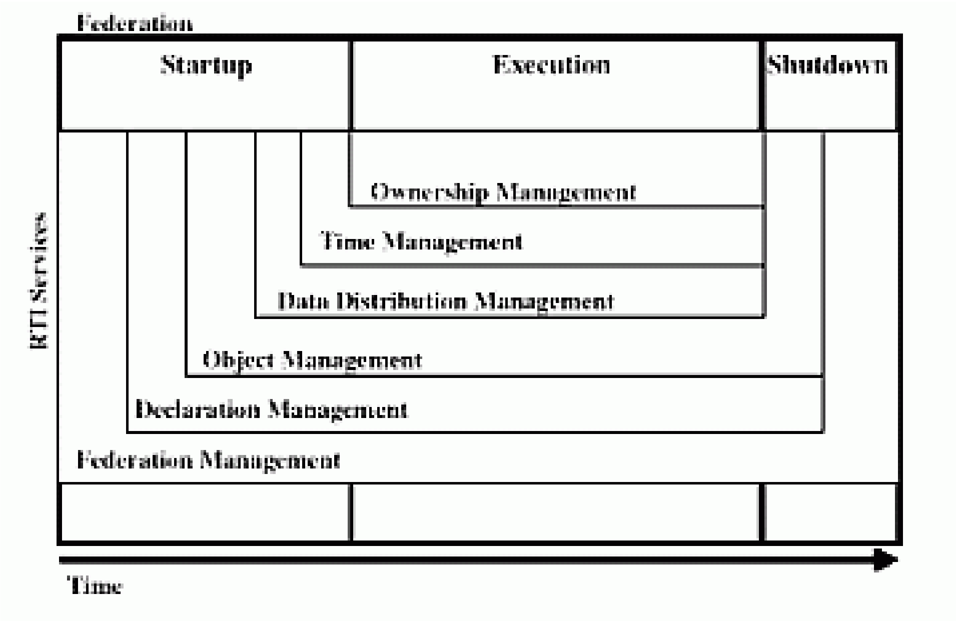
- Join virtual-level simulators
  - Tanks, Aircraft, Helicopters, Soldiers, Vehicles
- Provide common digital environment
  - Terrain, Features, Vehicles, Lighting
- Support fair-fight between different types of simulators
  - Detection, Engagement

- Run in real-time
    - No perceptible lag for users

Distributed Interactive Simulation (DIS) is best known for a catalog of message protocols, but it also included extensive descriptions of how to make simulators work together and how to manage a DIS-driven training event.



RTI Service Life-Cycle

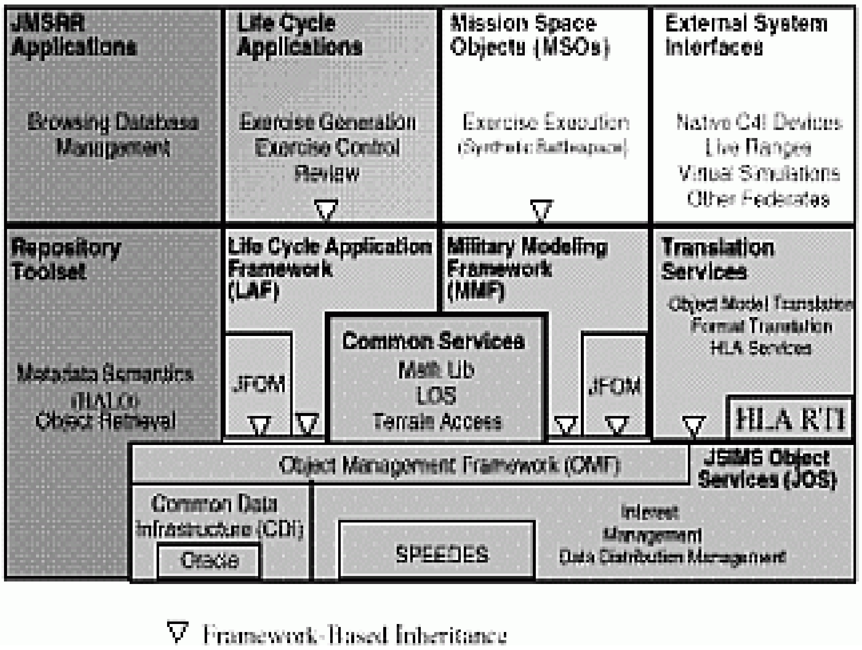


The six categories of RTI services are useful during specific phases of an exercise. This diagram emphasizes that most services are needed for both federation startup and execution.

JSIMS Architecture v2.0

This architecture picture further refines the JSIMS architecture. Oracle has been selected as the Common Data Interchange product. The HLA RTI has been relegated to the role of a Translation Service for communicating with non-JSIMS simulations and other computer systems. Framework-Based Inheritance is specifically identified.

This provides base classes from which high layers inherit capability. This is unique from the traditional Application Programming Interface (API) in which high layers call functions in lower layers.

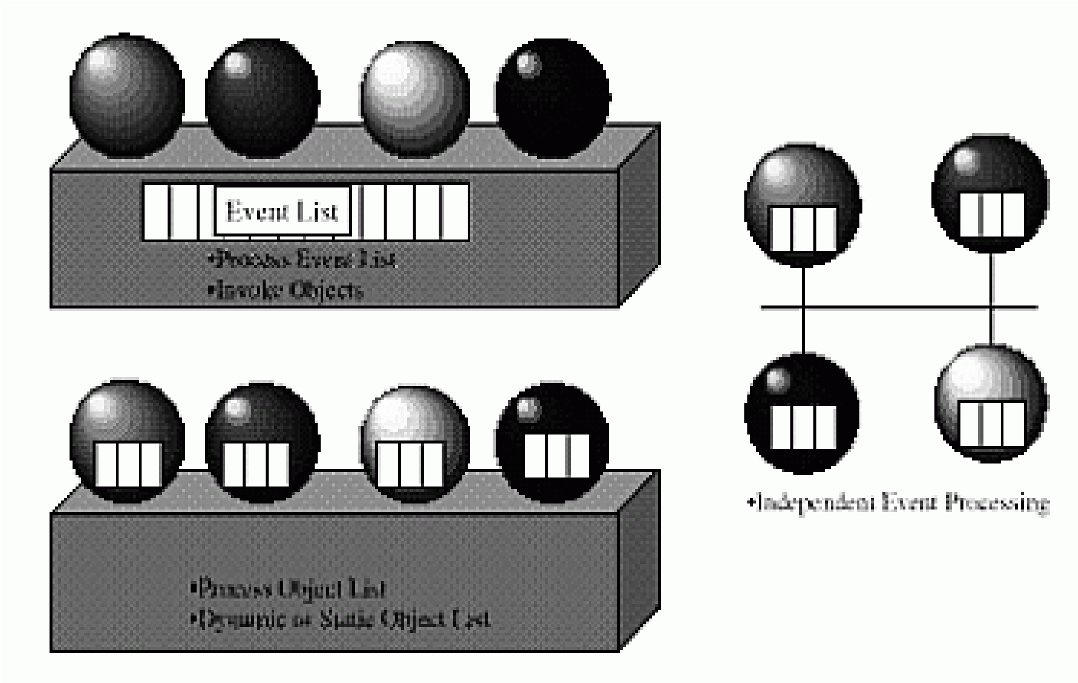


Event Management Options

Events may be owned and processed by either the infrastructure or the objects themselves. However, if the objects are synchronized in time the infrastructure must be able to influence the time at which events are processed.

1. A master event list can be held by the infrastructure. The infrastructure has primary thread of control and invokes objects as specified in the event being processed.

2. The event lists can be owned directly by the objects that will execute them. However, the infrastructure must have some control over when the events are processed. This can be accomplished by allowing the infrastructure to process through a list of



objects, telling them what time they are allowed to process to. The Object list can be static (everyone all the time), or it can be based on a registration process in which objects schedule themselves for some future processing.

3. Each object can be a separate thread of control and own its own events. These objects may be unconstrained by time, or they may be tracking time through the exchange of time events.

4. Hybrid combinations of all these methods can and are used in real systems.

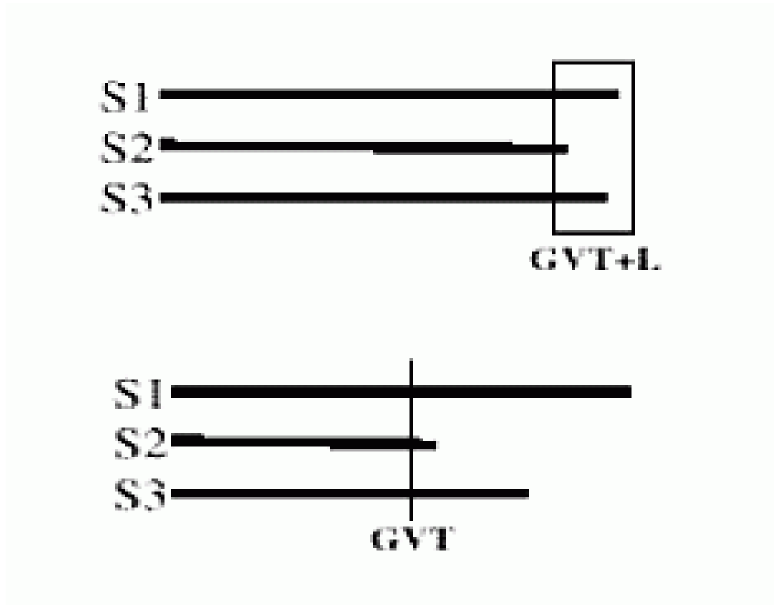
Time Management in DIS

- Paced (real-time or scaled real-time) execution
- Simulation time essentially the same as wallclock time (plus an offset)
- Autonomous simulation nodes, each broadcasts state changes (PDUs) as they occur
- Receiver determines information relevant to it
- Messages typically processed as they arrive (receive order)
- Message latency tolerance bounded by limitations in human perception (typically, up to 100 millisecond delay can be tolerated)
- Receiver may compensate for delay due to message latency.

A member of DIS exercise will receive event messages on the network that have a simulation time stamp slightly in the past of their own system clock. This delay can make the distributed world disjoint. Network delivery speed is an essential factor in ensuring that messages are delivered fast enough that they can be meaningfully applied in the other systems.

Distributed Time Management

We will explore mechanisms to handle time management in two very different manners. These are Conservative and Optimistic synchronization.

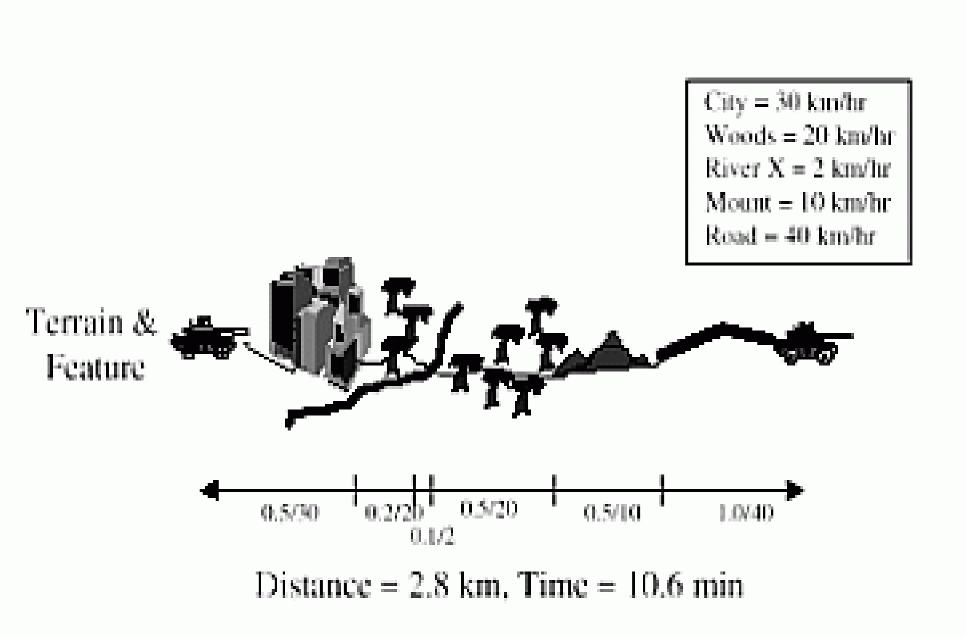


- Conservative
  - Keep all simulations and events synchronized at all times
  - Regulate progress of all by rate of slowest
- Optimistic
  - Allow independent progression, but insist on event synchronization
  - Allow "run forward’ and "roll back"

Simple Movement

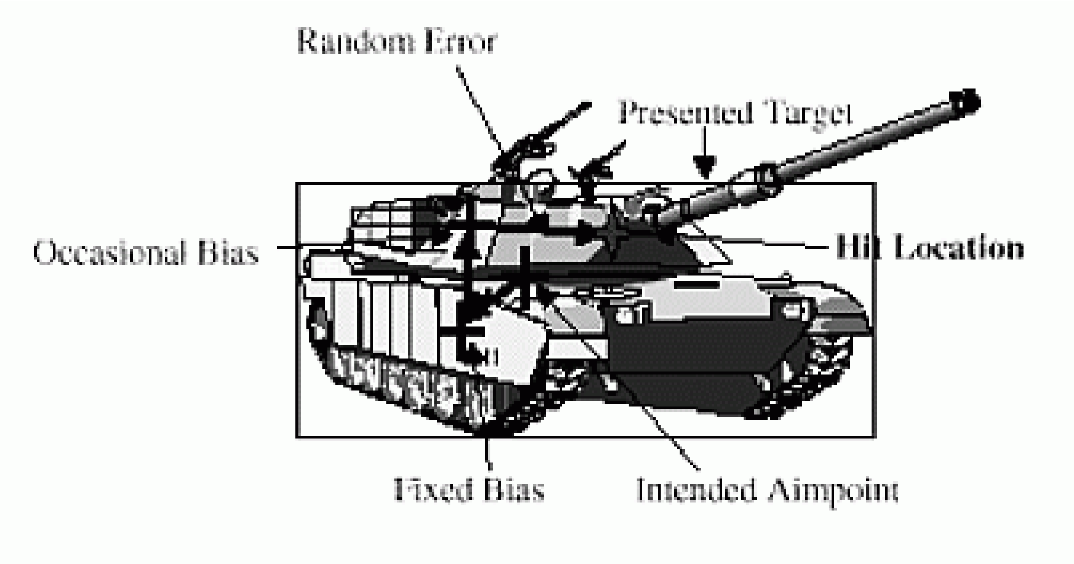
Adding detail to the terrain that the vehicles move on will impose additional time to cover the distance, and will cause the vehicle to deviate from the straight line route. These deviations require the addition of logic to insure that the objects remain oriented toward their original objective. This is the beginning of the imposition of behavioral modeling in the physical modeling domain.

It is actually very difficult to make clear divisions between physical, environmental, and behavioral models. The current practice is to treat certain factors of each as if they actually belong to the other class. For example, though all decisions for traversing a route are behavioral/cognitive tasks, these are so essential to movement modeling that they are often included in the physical model. At each boundary a decision has to be made about the category for every representation - no matter how arbitrary this decision may be.



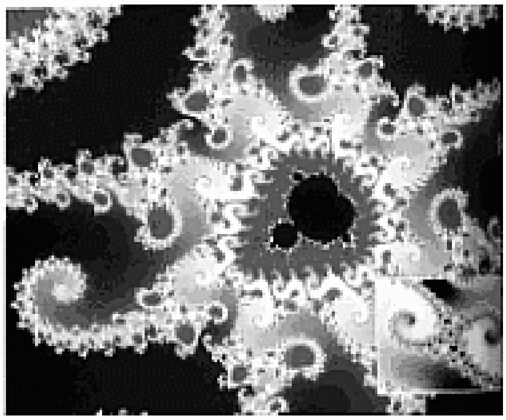
ModSAF Weapon Accuracy

- Fixed Bias - Constant ordnance error
- Occasional Bias - Ordnance error under condition
- Random Error - Round to round dispersion



The accuracy of a weapon in ModSAF plays a part in the PK that is assigned to the engagement. Each target provides some presented area to the shooter. The shooter places an aim point on this vehicle and fires the weapon. However, the engagement is modified by a fixed bias for the type of weapon used, an occasional bias for the variations caused by the positioning of the two vehicles, and a random bias for the round-to-round differences found in any form of munition.

Chaos Theory



- **Traditional Thinking:** Minor Changes in Input Yields Minor Changes in Outcome
- **Chaos Theory:** Minor Changes in Input Yields Major Changes in Outcome
- Threatens to invalidate predictive power of simulations. "Very small differences in a system result in very large differences in the behavior of the system."[1]

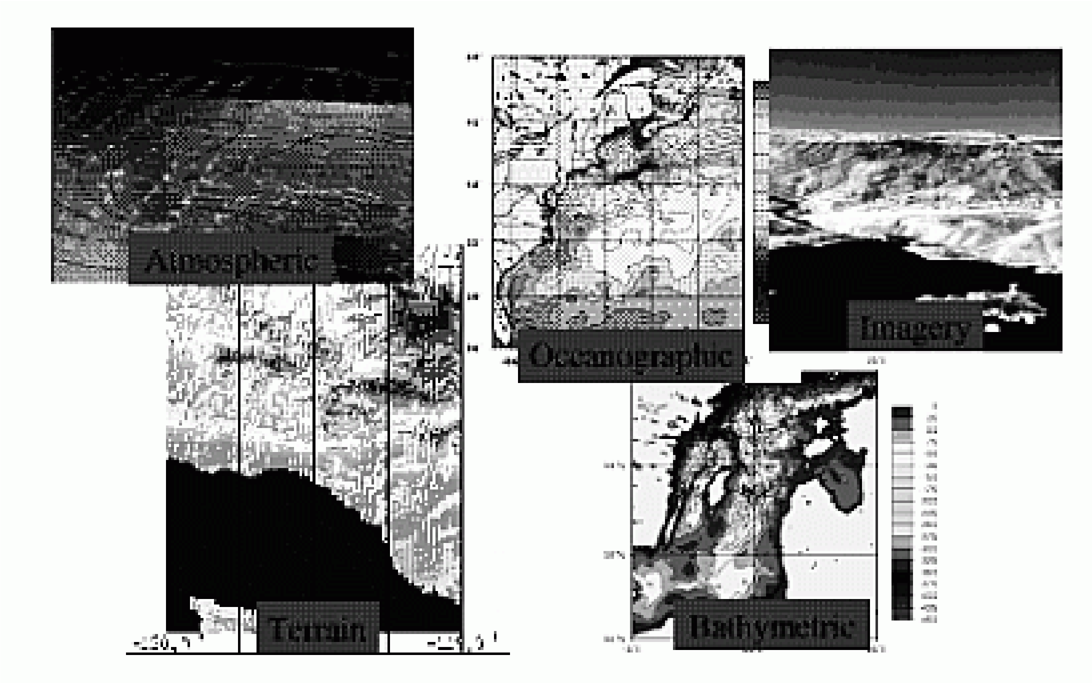
As modelers we have attempted to insure that input data and model parameters are as accurate as possible. However, we have also assumed that minor errors or variations in this data will only have minor effects on the outcome of the simulation runs. Following the explosion in Chaos Theory in the late 1980’s simulation scientists began to question whether our input data could be chaotic variables.

If Chaos Theory is applicable to military simulations this means that minor variations in input data can create huge changes in the outcome of the simulation. At the extreme, this could mean that small changes in the placement of units, slightly different effectiveness variables, and minor changes in the timing at which orders are entered can totally change the outcome of the war.

Since simulations are very complex systems with many interactions between the variables involved, it is possible for two adjacent values to be transformed along totally divergent paths. In practice we believe that most variations dampen out, but the potential does exist for chaotic behavior. This has potentially dire implications on the ability of a model to do analysis and prediction.

Environmental Data

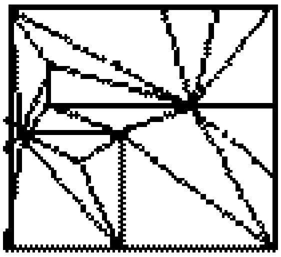
An essential step in environmental modeling is the collection or creation of data to represent the specific type of environment required. This includes information about the terrain, atmosphere, ocean surface, ocean, floor, vegetation, imagery, etc. The list is extensive and the storage requirements are



large.

Luckily, this type of data is also essential for modern war fighting. Therefore, the information is available from the same sources that provide it to combat systems. All libraries of data usually go through a transformation, enhancement, or optimization process to prepare them for use in simulations, image generators, and other tools.

TIN

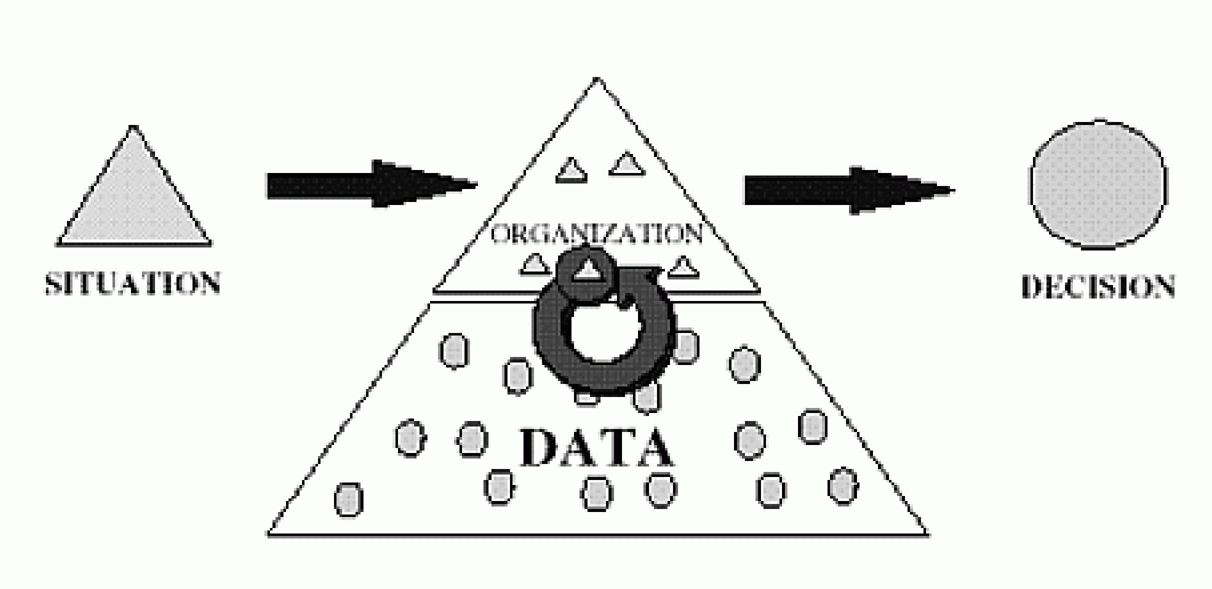


- Triangulated Irregular Networks
- Flexible terrain fitting algorithms
- Reduce sampled data volume, retain higher data accuracy
- Variable resolution based on sampled terrain and features

Triangulated Irregular Networks (TIN) are a very efficient and flexible method for storing environmental data. These allow the data modeler to represent the environment at different levels of detail and to transform data between these layers.

When fitting polygons to an underlying data source either of the tessellating shapes described earlier can be used (triangle, rectangle, and hexagon). However, the triangle is unique in its ability to subdivide itself, which the hexagon can not do. Three points are also guaranteed to lie in a definable spatial plane, which is not true for either the rectangle or the hexagon.

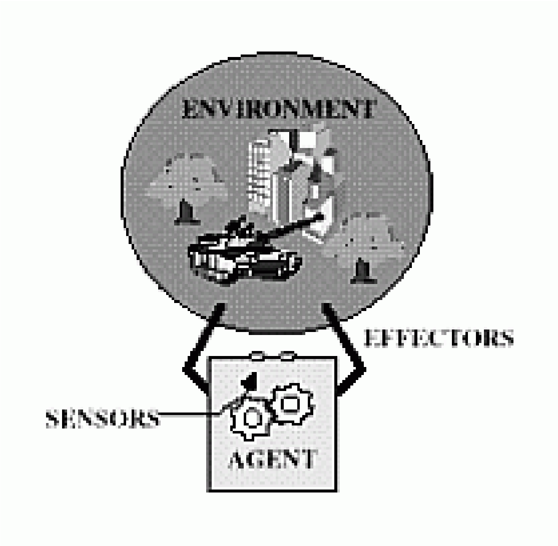
Behavioral Modeling



The generation of a decision usually begins with a situation. This enters the model where it is processed and an appropriate set of responses are located.

From these a single decision/action is selected and propagated as the behavior that will be exhibited.

Basic Intelligent Agent



- Encapsulates behavior
- Behavior model is distinct from environment
- Perceives the environment through sensors
- Acts on the environment through effectors

Agents are an encapsulating technique for organizing the behavioral models interactions and effects on the outside world.

- "anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors."[2]
- "a self-contained software element responsible for executing part of a programmatic process, usually in a distributed environment."[3]
- "communicates correctly with peer entities by exchanging messages in an agent communication language."[4]
- "makes use of non-procedural process information - knowledge - defined in and accessed from a knowledge base, by means of inference mechanisms."[5]

FSM in Military Training

Vehicle Towing Operation

Start

Enter MOVE State

Move

MOVE until distance closed

Hitching

First switch out of move

Enter HITCH State

End HITCH, enter MOVE

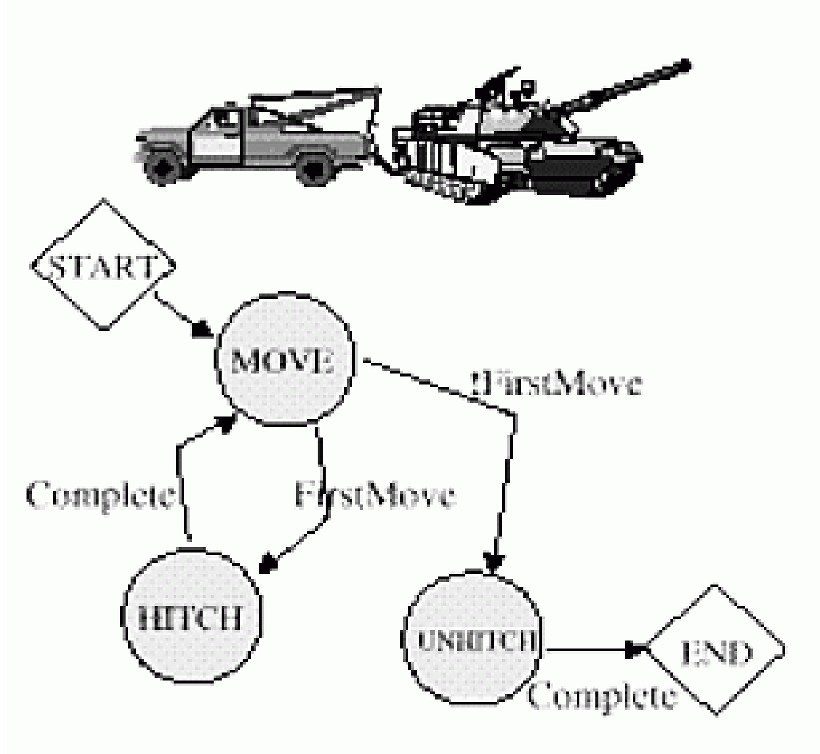
Unhitching

After First switch out of move

Enter UNHITCH State

End

End UNHITCH



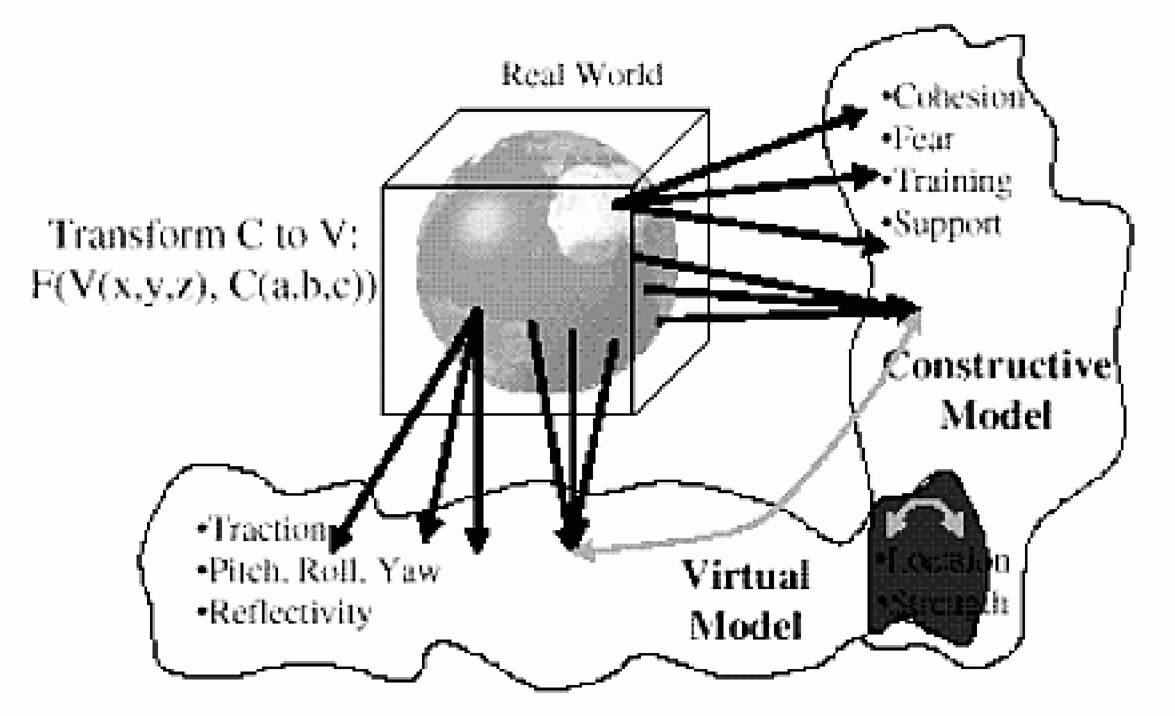
FSM Complete

This is the FSM used within the CCTT system to represent the recovery of a damaged tank by a "tow-tank."

The two-tank first enters a "Move" state which is used to bring the vehicle into the general vicinity of the damaged tank to be rescued. When the tow-tank is within range it switches into the "Hitch" state. This includes all of the backing and positioning to align the tow equipment with the tow point on the damaged tank. This part of the FSM may become very complicated to account to the presence of terrain and other vehicles that interfere with the operation. However, the FSM was selected for the problem specifically because this complexity could be contained and not interfere with the other phases of the operation. Following the "Hitch" state, the tow-tank returns to the "Move" state to drag the damaged tank to a location suitable for repair. Upon arrival at the repair station, the tow-tank enters the "Unhitch" state which models the processes of getting the damaged tank off of the towing equipment and positioned properly in the repair area.

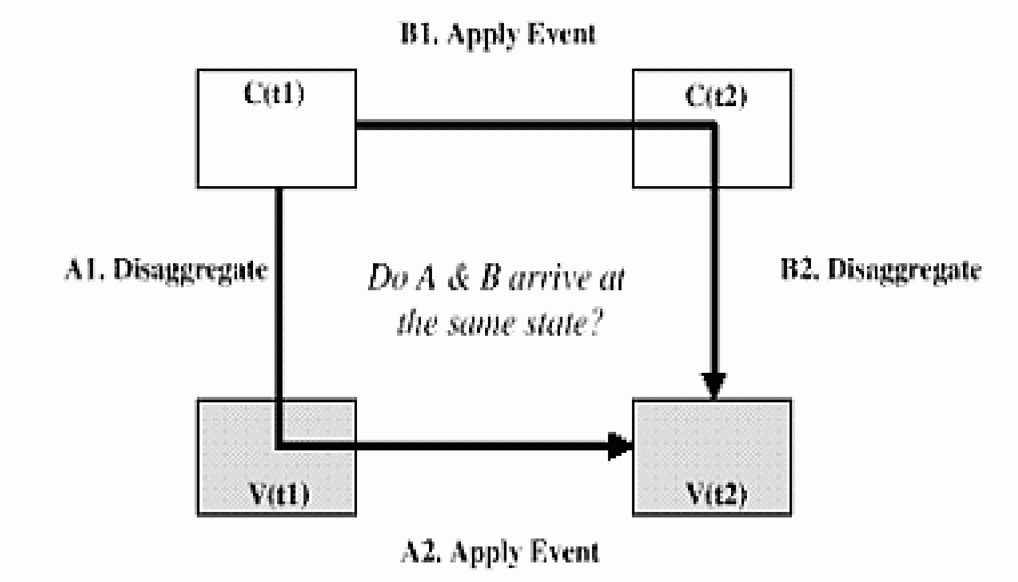
Two World Representations

Current MRM prototypes have only touched on the easiest variables to transform from one domain to the other. A small set of variables exist in both models and can be translated more-or-less directly from one form to the other. However, a much larger set of variables are represented very uniquely in each model. When these are transformed to another model you must establish the relationship between them. This relationship may not be obvious or traceable in either model, but may require tracing the variables back to their original source and identifying the relationships between them in that domain (usually in the real world).

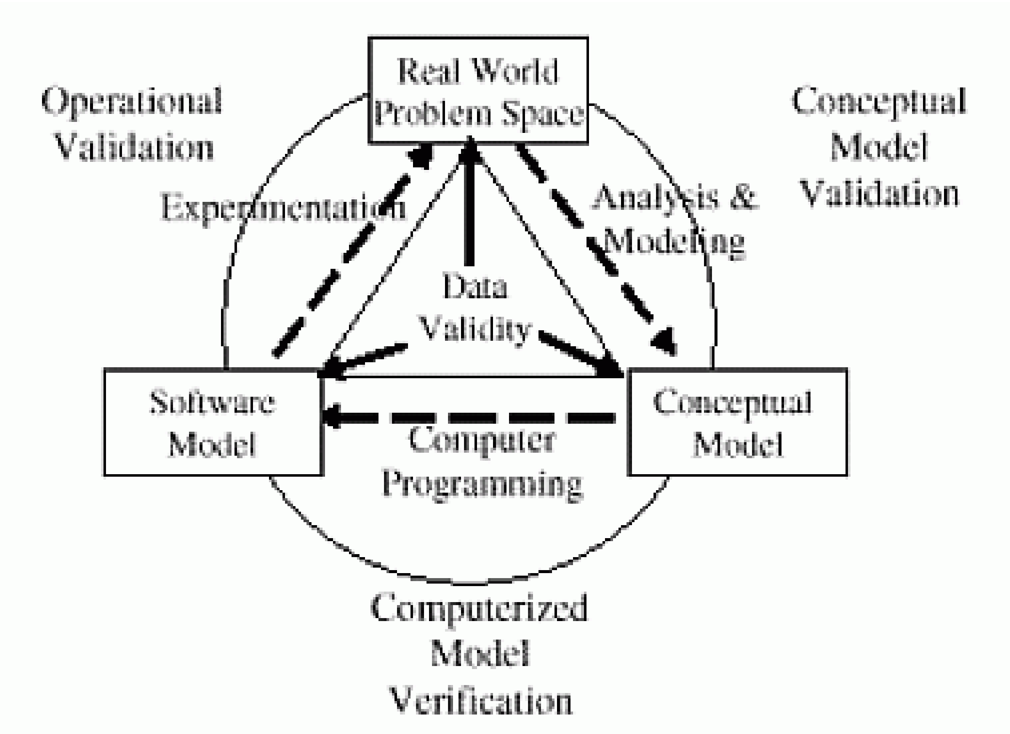


Multi-Resolution Model Consistency

Paul Davis of RAND Corp. pointed out that consistency between the aggregate and entity level view of the world could be tested by running experiments in which one applied the disaggregation operation and then the combat event, then compared this to the result of applying the combat event at the aggregate level followed by the disaggregation operation.



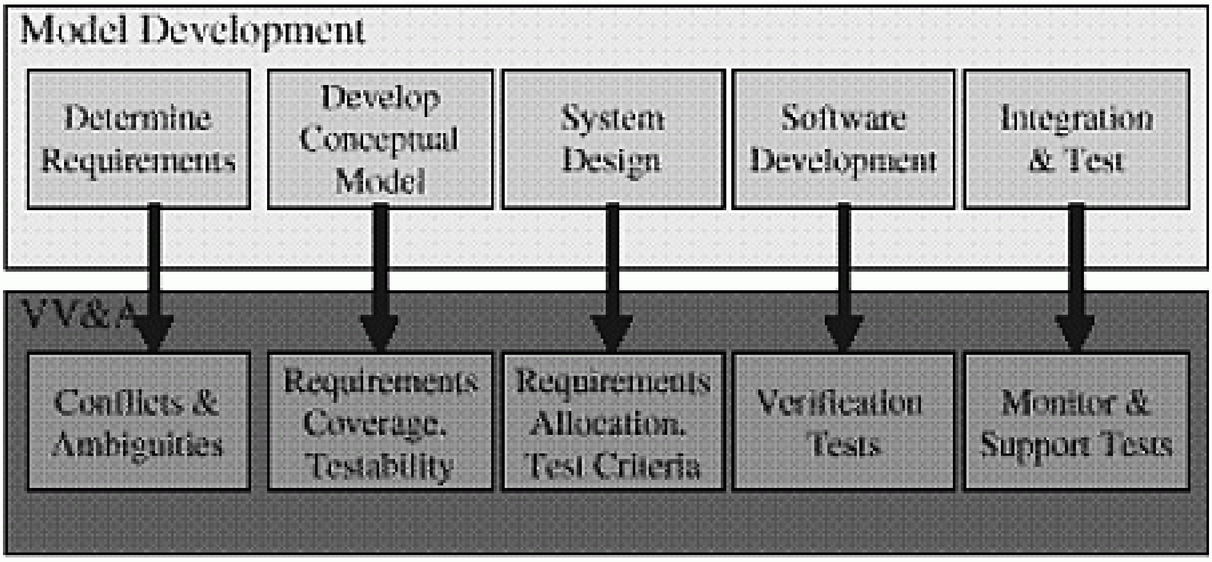
Verification & Validation



The transformation between each of these representations of the system presents a potential for error. Conceptual model validation is conducted to insure that the creation of the conceptual model captured all of the important aspects of the real system. It also insures the proper balance and interactions between objects in the conceptual model. Computerized model verification is performed to insure that the software model is an accurate representation of the conceptual model. The creation of software is a very error prone activity and it is likely that the ideas so carefully crafted in the conceptual model were not accurately captured in software. Operational validation is conducted to compare the software model to the real system. This is the final check to see that the final product does behave in a fashion that is similar and representative of the real world.

Collaborative VV&A

For each step in the development of a simulation system there is a corresponding activity in the VV&A realm. VV&A is not meant as a surprise test that must be passed at the end of a project. It is a continuous guiding light to help lead the developers to a valid solution to the problem presented. Deviations from the appropriate solution should be detected and addressed early by the VV&A practitioners and model developers.



DiSTI Courses

- DiSTI offers courses on:
- Military Simulation Techniques & Technology
- Simulation Foundations
- High Level Architecture
- Distributed Interactive Simulation
- Real-time Platform Reference Federation
- SEDRIS

This and other DiSTI courses are taught regularly throughout the year. They are also available for in-house presentation. Customized versions have been created to focus on specific areas of military simulation. Updated information, schedules and registration forms are available at http://www.simulation.com

---

References:

1. **Henri Poincare, late 1800s.**
2. **Russell & Norvig, Artificial Intelligence: A Modern Approach.**
3. **Guifoyle & Warner, Intelligent Agents: The New Revolution in Software.**
4. **Riecken, Communications of the ACM 37(7), 1994.**
5. **Guifoyle & Warner, Intelligent Agents: The New Revolution in Software.**

ROGER SMITH is the instructor for this course. He has been collecting and refining the content of the course since 1996. The course provides the most current and important technologies for developing simulation systems. The author is an Award Winning simulation developer who is intimately involved in designing, developing, and fielding military simulations. He has contributed to JSIMS, WARSIM, NASM, J-SIGSIM, TACSIM, AWSIM, ALBAM, and F-16 flight simulators. He is a leader in the simulation industry, serving as the Chairman for the ACM Special Interest Group on Simulation, the General Chairman for the Electronic Conferences on Training Simulation, and on the Editorial Boards of *ACM Transactions on Modeling and Computer Simulation and the International Journal of Computer Simulation, Modeling, and Analysis*. He is a Technical Director for STAC Technologies, an Adjunct Professor at Florida Institute of Technology, and a regular lecturer at Georgia Tech and other universities. Address: 424 Research Parkway, Suite 380, Orlando, Florida 32826. Tel.: (407) 206-3390. E-mail: registra@simulation.com.

BACK TO TOP

THEORY OF MODELING AND SIMULATION

by Bernard P. Zeigler, Herbert Praehofer, Tag Gon Kim

2nd Edition, Academic Press, 2000, ISBN: 0127784551

Given the many advances in modeling and simulation in the last decades, the need for a widely accepted framework and theoretical foundation is becoming increasingly necessary. Methods of modeling and simulation are fragmented across disciplines making it difficult to re-use ideas from other disciplines and work collaboratively in multidisciplinary teams. Model building and simulation is becoming easier and faster through implementation of advances in software and hardware. However, difficult and fundamental issues such as model credibility and interoperation have received less attention. These issues are now addressed under the impetus of the High Level Architecture (HLA) standard mandated by the U.S. DoD for all contractors and agencies.

This book concentrates on integrating the continuous and discrete paradigms for modeling and simulation. A second major theme is that of distributed simulation and its potential to support the co-existence of multiple formalisms in multiple model components. Prominent throughout are the fundamental concepts of modular and hierarchical model composition. These key ideas underlie a sound methodology for construction of complex system models.

The book presents a rigorous mathematical foundation for modeling and simulation. It provides a comprehensive framework for integrating various simulation approaches employed in practice, including such popular modeling methods as cellular automata, chaotic systems, hierarchical block diagrams, and Petri Nets. A unifying concept, called the DEVS Bus, enables models to be transparently mapped into the Discrete Event System Specification (DEVS). The book shows how to construct computationally efficient, object-oriented simulations of DEVS models on parallel and distributed environments. In designing integrative simulations, whether or not they are HLA compliant, this book provides the foundation to understand, simplify and successfully accomplish the task.

MODELING HUMAN AND ORGANIZATIONAL BEHAVIOR: APPLICATION TO MILITARY SIMULATIONS

Editors: Anne S. Mavor, Richard W. Pew

National Academy Press, 1999, ISBN: 0309060966. Hardcover - 432 pages.

This book presents a comprehensive treatment of the role of the human and the organization in military simulations. The issue of representing human behavior is treated from the perspective of the psychological and organizational sciences. After a thorough examination of the current military models, simulations and requirements, the book focuses on integrative architectures for modeling the individual combatant, followed by separate chapters on attention and multitasking, memory and learning, human decision making in the framework of utility theory, models of situation awareness and enabling technologies for their implementation, the role of planning in tactical decision making, and the issue of modeling internal and external moderators of human behavior.

The focus of the tenth chapter is on modeling of behavior at the unit level, examining prior work, organizational unit-level modeling, languages and frameworks. It is followed by a chapter on information warfare, discussing models of information diffusion, models of belief formation and the role of communications technology. The final chapters consider the need for situation-specific modeling, prescribe a methodology and a framework for developing human behavior representations, and provide recommendations for infrastructure and information exchange.

The book is a valuable reference for simulation designers and system engineers.

HANDBOOK OF SIMULATOR-BASED TRAINING

by Eric Farmer (Ed.), Johan Reimersma, Jan Moraal, Peter Jorna

Ashgate Publishing Company, 1999, ISBN: 0754611876.

The rapidly expanding area of military modeling and simulation supports decision making and planning, design of systems, weapons and infrastructure. This particular book treats the third most important area of modeling and simulation – training. It starts with thorough analysis of training needs, covering mission analysis, task analysis, trainee and training analysis. The second section of the book treats the issue of training program design, examining current practices, principles of training and instruction, sequencing of training objectives, specification of training activities and scenarios, methodology of design and optimization of training programs. In the third section the authors introduce the problem of training media specification and treat technical issues such as databases and models, human-simulator interfaces, visual cueing and image systems, haptic, kinaesthetic and vestibular cueing, and finally, the methodology for training media specification. The final section of the book is devoted to training evaluation, covering the topics of performance measurement, workload measurement, and team performance. In the concluding part the authors outline the trends in using simulators for training.

The primary audience for this book is the community of managers and experts involved in training operators. It can also serve as useful reference for designers of training simulators.

CREATING COMPUTER SIMULATION SYSTEMS:

An Introduction to the High Level Architecture

by Frederick Kuhl, Richard Weatherly, Judith Dahmann

Prentice Hall, 1999, ISBN: 0130225118. - 212 pages.

Given the increasing importance of simulations in nearly all aspects of life, the authors find that combining existing systems is much more efficient than building newer, more complex replacements. Whether the interest is in business, the military, or entertainment or is even more general, the book shows how to use the new standard for building and integrating modular simulation components and systems. The HLA, adopted by the U.S. Department of Defense, has been years in the making and recently came ahead of its competitors to grab the attention of engineers and designers worldwide.

The book and the accompanying CD-ROM set contain an overview of the rationale and development of the HLA; a Windows-compatible implementation of the HLA Runtime Infrastructure (including test software). It allows the reader to understand in-depth the reasons for the definition of the HLA and its development, how it came to be, how the HLA has been promoted as an architecture, and why it has succeeded. Of course, it provides an overview of the HLA examining it as a software architecture, its large pieces, and chief functions; an extended, integrated tutorial that demonstrates its power and applicability to real-world problems; advanced topics and exercises; and well-thought-out programming examples in text and on disk.

The book is well-indexed and may serve as a guide for managers, technicians, programmers, and anyone else working on building simulations.

HANDBOOK OF SIMULATION:

Principles, Methodology, Advances, Applications, and Practice

edited by Jerry Banks

John Wiley & Sons, 1998, ISBN: 0471134031. Hardcover - 864 pages.

Simulation modeling is one of the most powerful techniques available for studying large and complex systems. This book is the first ever to bring together the top 30 international experts on simulation from both industry and academia. All aspects of simulation are covered, as well as the latest simulation techniques. Most importantly, the book walks the reader through the various industries that use simulation and explains what is used, how it is used, and why.

This book provides a reference to important topics in simulation of discrete- event systems. Contributors come from academia, industry, and software development. Material is arranged in sections on principles, methodology, recent advances, application areas, and the practice of simulation. Topics include object-oriented simulation, software for simulation, simulation modeling, and experimental design. For readers with good background in calculus based statistics, this is a good reference book.

Applications explored are in fields such as transportation, healthcare, and the military. Includes guidelines for project management, as well as a list of software vendors. The book is co-published by Engineering and Management Press.

ADVANCES IN MISSILE GUIDANCE THEORY

by Joseph Z. Ben-Asher, Isaac Yaesh

AIAA, 1998, ISBN 1-56347-275-9.

This book about terminal guidance of intercepting missiles is oriented toward practicing engineers and engineering students. It contains a variety of newly developed guidance methods based on linear quadratic optimization problems. This application-oriented book applies widely used and thoroughly developed theories such LQ and H-infinity to missile guidance. The main theme is to systematically analyze guidance problems with increasing complexity. Numerous examples help the reader to gain greater understanding of the relative merits and shortcomings of the various methods. Both the analytical derivations and the numerical computations of the examples are carried out with MATLAB Companion Software: The authors have developed a set of MATLAB M-files that are available on a diskette bound into the book.

CONTROL OF SPACECRAFT AND AIRCRAFT

by Arthur E. Bryson, Jr.

Princeton University Press, 1994, ISBN 0-691-08782-2.

This text provides an overview and summary of flight control, focusing on the best possible control of spacecraft and aircraft, i.e., the limits of control. The minimum output error responses of controlled vehicles to specified initial conditions, output commands, and disturbances are determined with specified limits on control authority. These are determined using the linear-quadratic regulator (LQR) method of feedback control synthesis with full-state feedback. An emphasis on modeling is also included for the design of control systems. The book includes a set of MATLAB M-files in companion software

---

MATHWORKS

http://www-europe.mathworks.com

Initial information MATLAB is given in this volume to allow to present next the Simulink package and the Flight Dynamics Toolbox, providing for rapid simulation-based design. MATLAB is the foundation for all the MathWorks products. Here we would like to discus products of MathWorks related to the simulation, especially Code Generation tools and Dynamic System Simulation.

Code Generation and Rapid Prototyping

The MathWorks code generation tools make it easy to explore real-world system behavior from the prototyping stage to implementation. *Real-Time Workshop* and *Stateflow Coder* generate highly efficient code directly from Simulink models and Stateflow diagrams. The generated code can be used to test and validate designs in a real-time environment, and make the necessary design changes before committing designs to production. Using simple point-and-click interactions, the user can generate code that can be implemented quickly without lengthy hand-coding and debugging. Real-Time Workshop and Stateflow Coder automate compiling, linking, and downloading executables onto the target processor providing fast and easy access to real-time targets. By automating the process of creating real-time executables, these tools give an efficient and reliable way to test, evaluate, and iterate your designs in a real-time environment.

Real-Time Workshop, the code generator for Simulink, generates efficient, optimized C and Ada code directly from Simulink models. Supporting discrete-time, multirate, and hybrid systems, Real-Time Workshop makes it easy to evaluate system models on a wide range of computer platforms and real-time environments.

Stateflow Coder, the standalone code generator for Stateflow, automatically generates C code from Stateflow diagrams. Code generated by Stateflow Coder can be used independently or combined with code from Real-Time Workshop.

Real-Time Windows Target, allows to use a PC as a standalone, self-hosted target for running Simulink models interactively in real time. Real-Time Windows Target supports direct I/O, providing real-time interaction with your model, making it an easy-to-use, low-cost target environment for rapid prototyping and hardware-in-the-loop simulation.

xPC Target allows to add I/O blocks to Simulink block diagrams, generate code with Real-Time Workshop, and download the code to a second PC that runs the xPC target real-time kernel. xPC Target is ideal for rapid prototyping and hardware-in-the-loop testing of control and DSP systems. It enables the user to execute models in real time on standard PC hardware.

By combining the MathWorks code generation tools with hardware and software from leading real-time systems vendors, the user can quickly and easily perform rapid prototyping, hardware-in-the-loop (HIL) simulation, and real-time simulation and analysis of your designs. Real-Time Workshop code can be configured for a variety of real-time operating systems, off-the-shelf boards, and proprietary hardware.

The MathWorks products for control design enable the user to make changes to a block diagram, generate code, and evaluate results on target hardware within minutes. For turnkey rapid prototyping solutions you can take advantage of solutions available from partnerships between The MathWorks and leading control design tools:

- *dSPACE Control Development System:* A total development environment for rapid control prototyping and hardware-in-the-loop simulation;
- *WinCon:* Allows you to run Real-Time Workshop code independently on a PC;
- *World Up:* Creating and controlling 3-D interactive worlds for real-time visualization;
- *ADI Real-Time Station:* Complete system solution for hardware-in-the loop simulation and prototyping.
- *Pi AutoSim:* Real-time simulator for testing automotive electronic control units (ECUs).
- *Opal-RT:* a rapid prototyping solution that supports real-time parallel/distributed execution of code generated by Real-Time Workshop running under the QNX operating system on Intel based target hardware.

Dynamic System Simulation

Simulink is a powerful graphical simulation tool for modeling nonlinear dynamic systems and developing control strategies. With support for linear, nonlinear, continuous-time, discrete-time, multirate, conditionally executed, and hybrid systems, Simulink lets you model and simulate virtually any type of real-world dynamic system. Using the powerful simulation capabilities in Simulink, the user can create models, evaluate designs, and correct design flaws before building prototypes.

Simulink provides a graphical simulation environment for modeling dynamic systems. It allows to build quickly block diagram models of dynamic systems. The Simulink block library contains over 100 blocks that allow to graphically represent a wide variety of system dynamics. The block library includes input signals, dynamic elements, algebraic and nonlinear functions, data display blocks, and more. Simulink blocks can be triggered, enabled, or disabled, allowing to include conditionally executed subsystems within your models.

FLIGHT DYNAMICS TOOLBOX – FDC 1.2

report by Marc Rauw

FDC is an abbreviation of Flight Dynamics and Control. The FDC toolbox for Matlab and Simulink makes it possible to analyze aircraft dynamics and flight control systems within one software environment on one PC or workstation. The toolbox has been set up around a general non-linear aircraft model which has been constructed in a modular way in order to provide maximal flexibility to the user. The model can be accessed by means of the graphical user-interface of Simulink. Other elements from the toolbox are analytical Matlab routines for extracting steady-state flight-conditions and determining linearized models around user-specified operating points, Simulink models of external atmospheric disturbances that affect the motions of the aircraft, radio-navigation models, models of the autopilot, and several help-utilities which simplify the handling of the systems. The package can be applied to a broad range of stability and control related problems by applying Matlab tools from other toolboxes to the systems from FDC 1.2. The FDC toolbox is particularly useful for the design and analysis of Automatic Flight Control Systems (AFCS). By giving the designer access to all models and tools required for AFCS design and analysis within one graphical Computer Assisted Control System Design (CACSD) environment the AFCS development cycle can be reduced considerably. The current version 1.2 of the FDC toolbox is an advanced proof of concept package which effectively demonstrates the general ideas behind the application of CACSD tools with a graphical user- interface to the AFCS design process.

MODELING AND SIMULATION TERMINOLOGY

MILITARY SIMULATION
TECHNIQUES & TECHNOLOGY

Introduction to Simulation

Definitions. Defines simulation, its applications, and the benefits derived from using the technology. Compares simulation to related activities in analysis and gaming.

DOD Overview. Explains the simulation perspective and categorization of the US Department of Defense.

Training, Gaming, and Analysis. Provides a general delineation between these three categories of simulation.

System Architectures

Components. Describes the fundamental components that are found in most military simulations.

Designs. Describes the basic differences between functional and object oriented designs for a simulation system.

Infrastructures. Emphasizes the importance of providing an infrastructure to support all simulation models, tools, and functionality.

Frameworks. Describes the newest implementation of an infrastructure in the forma of an object oriented framework from which simulation capability is inherited.

Interoperability

Dedicated. Interoperability initially meant constructing a dedicated method for joining two simulations for a specific purpose.

DIS. The virtual simulation community developed this method to allow vehicle simulators to interact in a small, consistent battlefield.

ALSP. The constructive, staff training community developed this method to allow specific simulation systems to interact with each other in a single joint training exercise.

HLA. This program was developed to replace and, to a degree, unify the virtual and constructive efforts at interoperability.

JSIMS. Though not labeled as an interoperability effort, this program is pressing for a higher degree of interoperability than have been achieved through any of the previous programs.

Event Management

Queuing. The primary method for executing simulations has been various forms of queues for ordering and releasing combat events.

Trees. Basic queues are being supplanted by techniques such as Red-Black and Splay trees which allow the simulation store, process, and review events more efficiently than their predecessors.

Event Ownership. Events can be owned and processed in different ways. Today's preference for object oriented representations leads to vehicle and unit ownership of events, rather than the previous techniques of managing them from a central executive.

Time Management

Universal. Single processor simulations made use of a single clocking mechanism to control all events in a simulation. This was extended to the idea of a "master clock" during initial distributed simulations, but is being replaced with more advanced techniques in current distributed simulation.

Synchronization. The "master clock" too often lead to poor performance and required a great deal of cross-simulation data exchange. Researchers in the Parallel Distributed Simulation community provided several techniques that are being used in today's training environment.

Conservative & Optimistic. The most notable time management techniques are conservative synchronization developed by Chandy, Misra, and Bryant, and optimistic synchronization (or Time Warp) developed by David Jefferson.

Real-time. In addition to being synchronized across a distributed computing environment, many of today's simulators must also perform as real-time systems. These operate under the additional duress of staying synchronized with the human or system clock perception of time.

Principles of Modeling

Science & Art. Simulation is currently a combination of scientific method and artistic expression. Learning to do this activity requires both formal education and watching experienced practitioners approach a problem.

Process. When a team of people undertake the development of a new simulation system they must follow a defined process. This is often re-invented for each project, but can better be derived from experience of others on previous projects.

Fundamentals. Some basic principles have been learned and relearned by members of the simulation community. These have universal application within the field and allow new developers to benefit from the mistakes and experiences of their predecessors.

Formalism. There has been some concentrated effort to define a formalism for simulation such that models and systems are provably correct. These also allow mathematical exploration of new ideas in simulation.

Physical Modeling

Object Interaction. Military object modeling is be divided into two pieces, the physical and the behavioral. Object interactions, which are often viewed as 'physics based', characterize the physical models.

Movement. Military objects are often very mobile and a great deal of effort can be given to the correct movement of ground, air, sea, and space vehicles across different forms of terrain or through various forms of ether.

Sensor Detection. Military object are also very eager to interact with each other in both peaceful and violent ways. But, before they can do this they must be able to perceive each other through the use of human and mechanical sensors.

Engagement. Encounters with objects of a different affiliation often require the application of combat engagement algorithms. There are a rich set of these available to the modeler, and new ones are continually being created.

Attrition. Object and unit attrition may be synonymous with engagement in the real world, but when implemented in a computer environment they must be separated to allow fair combat exchanges. Distributed simulation systems are more closely replicating real world activities than did their older functional/sequential ancestors, but the distinction between engagement and attrition are still important.

Communication. The modern battlefield is characterized as much by communication and information exchange as it is by movement and engagement. This dimension of the battlefield has been largely ignored in previous simulations, but is being addressed in the new systems under development today.

More. Activities on the battlefield are extremely rich and varied. The models described in this section represent some of the most fundamental and important, but they are only a small fraction of the detail that can be included in a model.

Behavioral Modeling

Perception. Military simulations have historically included very crude representations of human and group decision making. One of the first real needs for representing the human in the model was to create a unique perception of the battlefield for each group, unit, or individual.

Reaction. Battlefield objects or units need to be able to react realistically to various combat environments. These allow the simulation to handle many situations without the explicit intervention of a human operator.

Planning. Today we look for intelligent behavior from simulated objects. Once form of intelligence is found in allowing models to plan the details of a general operational combat order, or to formulate a method for extracting itself for a difficult situation.

Learning. Early reactive and planning models did not include the capability to learn from experience. Algorithms can be built which allow units to become more effective as they become more experienced. They also learn the best methods for operating on a specific battlefield or under specific conditions.

Artificial Intelligence. Behavioral modeling can benefit from the research and experience of the AI community. Techniques of value include: Intelligent Agents, Finite State Machines, Petri Nets, Expert and Knowledge-based Systems, Case Based Reasoning, Genetic Algorithms, Neural Networks, Constraint Satisfaction, Fuzzy Logic, and Adaptive Behavior. An introduction is given to each of these along with potential applications in the military environment.

Environmental Modeling

Terrain. Military objects are heavily dependent upon the environment in which they operate. The representation of terrain has been of primary concern because of its importance and the difficulty of managing the amount of data required. Triangulated Irregular Networks (TINs) are one of the newer techniques for managing this problem.

Atmosphere. The atmosphere plays an important role in modeling air, space, and electronic warfare. The effects of cloud cover, precipitation, daylight, ambient noise, electronic jamming, temperature, and wind can all have significant effects on battlefield activities.

Sea. The surface of the ocean is nearly as important to naval operations as is terrain to army operations. Sub-surface and ocean floor representations are also essential for submarine warfare and the employment of SONAR for vehicle detection and engagement.

Standards. Many representations of all of these environments have been developed.

Unfortunately, not all of these have been compatible and significant effort is being given to a common standard for supporting all simulations. Synthetic Environment Data Representation and Interchange Specification (SEDRIS) is the most prominent of these standardization efforts.

Multi-Resolution Modeling

Aggregation. Military commanders have always dealt with the battlefield in an aggregate form. This has carried forward into simulations which operate at this same level, omitting many of the details of specific battlefield objects and events.

Disaggregation. Recent efforts to join constructive and virtual simulations have required the implementation of techniques for cross the boundary between these two levels of representation. Disaggregation attempts to generate an entity level representation from the aggregate level by adding information. Conversely, aggregation attempts to create the constructive from the virtual by removing information.

Interoperability. It is commonly accepted that interoperability in these situations is best achieved though disaggregation to the lowest level of representation of the models involved. In any form the patchwork battlefield seldom supports the same level of interoperability across model levels as is found within models at the same level of resolution.

Inevitability. Models are abstractions of the real world generated to address a specific problem. Since all problems are not defined at the same level of physical representation, the models built to address them will be at different levels. The modeling an simulation problem domain is too rich to ever expect all models to operate at the same level. Multi-Resolution Modeling and techniques to provide interoperability among them are inevitable.

Verification, Validation, and Accreditation

Verification. Simulation systems and the models within them are conceptual representations of the real world. By their very nature these models are partially accurate and partially inaccurate. Therefore, it is essential that we be able to verify that the model constructed accurately represents the important parts of the real world we are try to study or emulate.

Validation. The conceptual model of the real world is converted into a software program. This conversion has the potential to introduce errors or inaccurately represent the conceptual model. Validation ensures that the software program accurately reflects the conceptual model.

Accreditation. Since all models only partially represent the real world, they all have limited application for training and analysis. Accreditation defines the domains and conditions under which a particular model can be reliably used.

VV&A Principles. The Department of Defense has established specific guidelines for conducting VV&A. Simulation researchers have also defined fundamental principles that are important for this activity.

Model Building Exercises

Modeling. In-class projects to explore the concepts presented in the lectures. These exercises demonstrate the process and product of modeling the real world.

Exploration. Students explore the questions involved in modeling. Learn to identify the objective of the system, interactions in the virtual world, objects that must be defined, and dynamic and static attributes of the objects.

Models and Infrastructure. Practical exercises demonstrate the power of a simulation infrastructure and how it is related to the models of the real world.

ACRONYMS

ADS	Advanced Distributed Simulation
AMG	Architecture Management Group
AMSO	Army Modeling and Simulation Office
API	Application Programmer Interface
ASOC	Air Sovereignty Operations Center
BBS	Brigade/Battalion Battle Simulation
C3	Command, Control and Communications
CAX	Computer Assisted (Aided) Exercise
CBS	Corps Battle Simulation
CCTT	Close Combat Tactical Trainer
CGF	Computer Generated Forces
CGI	Computer Graphic Interface / Common Gateway Interface

CONOPS	Concept of Operations
CORBA	Common Object Request Broker Architecture
COTS	Commercial off-the-shelf
CSSTSS	Combat Service Support Training Simulation System
CTDB	Compact Terrain Database Base
DARPA	Defense Advanced Research Projects Agency
DBMS	Data-Base Management System
DCOM	Distributed COM (Component Object Model)
DCS	Data Coding Standard
DIS	Distributed Interactive Simulation
DMSO	Defense Modeling and Simulation Office
DoD	Department of Defense
DOM	Document Object Model
DRM	Data Representation Model
EXCIMS	Executive Council for Modeling and Simulation
FEDEP	Federation Development and Execution Process
FIFO	First In First Out
FOM	Federation Object Model
FZD	Fire Zone Defense
GIAC	Graphics Input Aggregate Control
GVT	Global Virtual Time
HLA	High-Level Architecture
HOP	Hasty Occupy Position (algorithm)
HTTP	Hyper-Text Transfer Protocol
IDL	Interface Definition Language

IIOP	Inter-ORB Communication Protocol
IMACCS	Integrated Monitoring, Analysis and Control COTS System
IMT	Information Management Terminal
JDBC	Java Database Connectivity
JSIMS	Joint Simulation System
JTC	Joint Training Confederation
JTLS	Joint Theater Level Simulation
JWORB	Java Web Object Request Broker
LAN	Local Area Network
LOC	Line-Of-Code
M&S	Modeling and Simulation
MILES	Multiple Integrated Laser Engagement System
MoD	Ministry of Defense
ModSAF	Modular Semi-Automated Forces
MODSIM	Modular Simulation Language
MOSAIC	MOdels & Simulations: Army Integrated Catalog
MPP	Message Processor Program
MRM	Multi-Resolution Model
MSIAC	Modeling and Simulation Information Analysis Center
MSRR	(Army) Modeling and Simulation Resource Repository
MTWS	Marine Air Ground Task Force Tactical Warfare Simulation
NC3A	NATO's Command Control and Consultancy Agency
NMCC	National Military Command Center

NPAC	Northeast Parallel Architectures Center (Syracuse University)
NTC	(Army) National Training Center
ODBC	Open Database Connectivity
OLEDB	Object Linking and Embedding – Data Base
OMDT	Object Model Development Tool
OMG/DARPA	Object Management Group/ Defense Advanced Research Project Agency
OPLAN	Operations Plan
PIMS	Partnership Information Management System
PSS	Persistent State Service
RDBMS	Relational Data-Base Management System
RDF	Resource Description Format
RENAISSANCE	Reusable Network Architecture Interoperable Space Science, Analysis, Navigation, and Control Environment
RESA	Research, Evaluation, and Systems Analysis
RMI	Remote Method Invocation
SAM	Surface-to-Air Missile
SAMPEX	Solar Anomalous Magnetospheric Particle Explorer
SMTP	Simple Message Transfer Protocol
SQL	Structured Query Language
SSN	Space Surveillance Network
STF	SEDRIS Transmittal Format
TACSIM	Tactical Simulation
TIN	Triangulated Irregular Networks
TRP	Target Reference Point

TVR	Televirtual
UML	Unified Modeling Language
VICTORS	Variable Intensity Computerized Training System
VRML	Virtual Reality Modeling Language
VV&A	Verification, Validation and Accreditation
WAN	Wide Area Network
XML	Extensible Markup Language

MODELING AND SIMULATION RELATED WEB SITES

Defense Modeling, Simulation & Tactical Technology Information Analysis Center (DMSTTIAC)

ADS	www.ads.msrr.dmso.mil	Authoritative Data Sources [DMSO]
AEDC	www.arnold.af.mil	Arnold Engineering Development Center [AF]
AFAMS	www.afams.af.mil	Air Force Agency for Modeling & Simulation
AFCA	www.afca.scott.af.mil	Air Force Communications Agency
AFIT	www.afit.af.mil	Air Force Institute of Technology
AFSAA	www.afsaa.hq.af.mil	Air Force Studies Analyses Agency
AIR FORCE	www.af.mil	Air Force Home Page

Air Force MSRR	www.Afmsrr.afams.af.mil	Air Force M&S Resource Repository
ALSP	www.stricom.army.mil/hla/amg/PRODUCTS/ALSP	Aggregate Level Simulation Protocol
AMG	www.hla.dmso.mil/hla/amg	Architecture Management Group [DMSO – HLA]
APG	www.apg.army.mil	Aberdeen Proving Ground [ARMY]
ARDEC	www.pica.army.mil	Armament R,D, &E Center [ARMY]
ARDEC – DIS	www.Dis.pica.army.mil	Armament R,D, &E Center - DIS [ARMY]
ARL	www.arl.mil/	Army Research Laboratory
ARMY	www.army.mil	Army Homepage
ARMY MSRR	www.msrr.army.mil	Army M&S Resource Repository
ASA, RDA, (ASA, AL&T)	www.sarda.army.mil	Asst Secretary of the Army for Research, Development, &Acquisition [effective 16 Feb99 ASA, AL&T]
ASNE MSEA	www.Msea.afccc.af.mil	Air & Space Natural Environment MSEA

ASTT	www.astt.com	Advanced Simulation Technology Thrust [DARPA/JSIMS]
ATDNet	www.atd.net	Advanced Technology Demonstration Network
AWSIM	www.wg.hanscom.af.mil/AWSIMR/	Air Warfare Simulation [AF}
BMD SSC	www.jntf.osd.mil/bmdssc/	Ballistic Missile Defense Simulation Support Center
BMDO	www.jntf.osd.mil/	Ballistic Missile Defense Organization
BMDOLINK	www.acq.osd.mil/bmdo/bmdolink/html	BMDOLink
BMDO MSRR	www.jntf.osd.mil/bmdssc/	BMDO M&S Resource Repository
C4ISR DSC	www.dsc.osd.mil	[Joint] C4ISR Decision Support Center
C4ISR JBC	www.jbc.js.mil	C4ISR Joint Battle Center
C4ISR Model	www.diisa.mil/D8/html/c4isr.html	DISA C4ISR Model [Federation]
CFS	www.itsi.disa.mil/	Center For Standards [DISA JIEO]
CMMS	www.dmso.mil/projects/cmms/	Conceptual Model of the Mission Space [DSMO]

COMPASS	www.Compass.saic.com	Common Operation Modeling, Planning, &Simulation Strategy
DARPA	www.darpa.mil	Defense Advanced Research Projects Agency
DDR&E	www.dtic.mil/ddre	Director Defense Research & Engineering [OSD]
DEEM	www.dis.anl.gov/DEEM	Dynamic Environmental Effects Model
DEFENSLINK	www.defenselink.mil	DefenseLINK
DIAS	www.dis.anl.gov/DIAS	Dynamic Information Architecture System
DISA	www.disa.mil	EFENSE information Systems Agency
DISA C4ISR Model	www.disa.mil/D8/html/c4isr.html	DISA C4ISR Model [Federation]
DISA Standards	www.itsi.disa.mil/links.html www.disa.mil	DISA Standards Links DISA Standards Site Index
DMSO	www.dsmo.mil	Defense Modeling and Simulation Office

DMSO Projects>		

>M&S Awards	www.dmso.mil/dmso/projects/award	DMSO M&S Award Program
>ADS	www.Ads.msrr.dmso.mil	Authorities Data Sources
>CMMS	www.dmso.mil/dmso/projects/cmms	Conceptual Model of the Mission Space
>DAVIE	www.dmso.mil/projacts/davie	Data Verification Interactive Editor
>DS	www.dmso.mil/ds	Data Standardization
>DMSTTIAC	www.Dmsttiac.iitri.org	Defense Modeling, Simulation, & Tactical Technology Information Analysis Center
>HLA	www.Hla.dmso.mil/hla	High – Level [Simulation] Architecture
>HBR	www.dmso.mil/dmso/projects/hbr`	Human Behavior Representation
>MEL	www.Mel.dmso.mil	Master Environment Library
>MSOSA	www.msosa.mil.inter.net	Modeling & Simulation Operational Support Activity
>MSRR	www.msrr.dmso.mil	Modeling &Simulation Resource Repository
>SEDRIS	www.sedris.org	Synthetic Environment Data Representation & Interchange Specification
>UOB	www.dmso.mil/dmso/projects/uob	Unit Order of Battle
>TMPO	www.tmpo.nima.mil	Terrain Modeling Project Office

>VV&A	www.dmso.mil/dmso/projects/vva	Verification, Validation, and Accreditation

DMSTTIAC	www.Dmsttiac.iitri.org	Defense Modeling, Simulation, & Tactical Technology Information Analysis Center [DMSO/DISA/DTIC]
DREN	www.arl.mil/HPCMP/DREN	Defense Research & Engineering Network
DS	www.dmso.mil/projects/ds	Data Standardization [DMSO]
DSC	www.dsc.osd.mil	[Joint C4ISR] Decision Support Center
DTIC	www.dtic.mil	Defense Technical Information Center
E2DIS	www.Vader.nrl.navy.mil/e2dis_www/home.html	Environment Effects for Distributed Interactive Simulation
EADSIM	www.smdc.army/mil/eadsim.html	Extended Air Defense Simulation
EADTB	www.smdc.army.mil/eadtb~2.html	Extended Air Defense Test bed
EXCIM	www.dmso.mil/dmso/wrkgrps/excims	Execute Council on Modeling &Simulation [DMSO]
HBR	www.dmso.mil/dmso/projects/hbr`	Human Behavior Representation [DMSO]
HLA	www.Hla.dmso.mil/hla	High – Level Architecture [DMSO}

HPCMP	www.hpcm.dren.net	High Performance Computing Modernization Program
ISOC	www.isoc.org	Internet Society
IST	www.ist.ucf.edu	Institute for Simulation & Training, Univ of Central Florida
ISTI	www.Disa11.disa.atd.net	Information Systems Technology Insertion [DSA]
ITEA	www.itea.org	International Test &Evaluation Association
JADS JTF	www.jads.abq.com	Joint Advanced Distributed Simulation Joint Test Force
JASA	www.nawcwpns.navy.mil/~jasa	Joint Accreditation Support Activity
JBC	www.jbc.js.mil	Joint [C4ISR] Battle Center
JSC	www.dtic.mil/jsc	Joint Chief of Staff
JDBE	208.245.129.4	Joint Data Base Element
JEL	www.dtic.mil/doctrine/jel/index.html	Joint Electronic Library
JMASS	www.jmas.wpafb.af.mil	Joint Modeling and Simulation System
JSF	www.jast.mil/html/jst_homepage.htm	Joint Strike Fighter
JSIMS	www.jsims.mil	Joint Simulation System
JTA	www.acq-ref.navy.mil/narsoc/jta021	Joint Technical Architecture [DoD]

JTASC	www.jtasc.acom.mil	Joint Training, Analysis, and Simulation Center [USACOM]
JWARS	www.dtic.mil/jwars	Joint Warfare System
JWFC	www.jwfc.acom.mil	Joint War fighting Center [USACOM]
MARINES	www.usmc.mil	Marine Corps Homepage

M&S Master Plans>	
>DoD	www.dmso.mil/dmso/docslib/mspolicy/msmp
>Army	www.amso.army.mil/mstrplin
>Navy	www.nawcad.navy.mil/tems/references/html
>Air Force	www.afams.af.mil/webdocs/afmsmp/
>Marine Corps	www.dmso.mil/dmso/docslib/mspolicy/usmcplan/
>NATO	www.dmso.mil/dmso/dicslib/mspolicy/nato_msmp/

M&S Management Office>		
>DMSO	www.dsmo.mil	Defense Modeling and Simulation Office
>AMSO	www.amso.army.mil/	Army Model and Simulation Office
>NAVMSMO	www.Navmso.hq.navy.mil	Navy Modeling and Simulation Management Office
>XOC	204.34.204.77/	AF Directorate of Command &Control [has M&S mgmt]

>> AFMS	www.afams.af.mil	Air Force Agency for Modeling and Simulation

MEL	www.Mel.dmso.mil	Master Environment Library [DMSO]
MORS	www.mors.org	Military Operations Research Society
MOVES	www.moves.nps.navy.mil	Modeling, Virtual Environment, and Simulation
MSOSA	www.msosa.mil.inter.net	Modeling & Simulation Operational Support Activity [DMSO]

MSRR>	www.msrr.dmso.mil	Modeling and Resource Repository [DMSO]
>Air Force MSRR	www.afsaa.hq.af.mil/index.html	Air Force Studies &Analyses Agency
>Army MSRR	www.msrr.army.mil	Army M&S Resource Repository
>Navy MSRR	www.Navmsmo.hq.navy.mil/nmsiscat/	Navy M&S Information System (NMSYS) Catalog
>BMDO MSRR	www.jntf.osd.mil/bmdssc/	BMDO M&D Resource Repository
>MEL	www.Mel.dmso.mil	Master Environmental Library [DMSO]
>ADS	www.ads.msrr.dmso.mil	Authoritative Data Sources [DMSO}
>C4ISR DSC	www.dsc.osd.mil/	C4ISR Decision Support Center [Joint]
>DIA	On SIPRNET	Defense Intelligent Agency

MSOC	www.dmso.mil/SOC	DoD M&S Staff Officer Course
MSTP	www.mstp.quantico.usmc.mil/	MAGTF Staff Training Program Center
MSTTF	www.dmso.mil/dmso/wrkgrps/tf/msttf.html	M&S Terminology Task Force [Glossary]
MSWG	www.dmso.mil/dmso/wrkgrps/mswg/	M&S Working Group [DMSO]
NASA	www.nasa.gov	National Aeronautics & Space Administration
NASA SW IV&V	www.ivv.nasa.gov	NASA Software Independent Verification & Validation Facility
NASM	www.nasm.hanscom.af.mil/NASM	National Air &Space (Warfare) Model [AF]
NAVSMO	www.Navsmo.hq.navy.mil	Navy Modeling & Simulation Management Office
NAVY	www.navy.mil	Navy Homepage
Navy MSRR	www.navmsmo.hq.navy.mil/nmsiscat/	Navy M&S Information System (NMSIS) catalog

NAVY TEMS	www.nawcad.navy.mil/tems	Navy Test & Evaluation Modeling & Simulation
NAWCTSD	www.ntsc.navy.mil	Naval Air Warfare Center, Training System dev9ision
NIMA	www.nima.mil	National Imagery and Mapping Agency
NIST	www.nist.gov	National Institute of Standards and Technology
NPSNET	www.npsnet.nps.navy.mil	Naval Postgraduate School NPSNET Research Group
NRL	www.nrl.navy.mil	Naval research laboratory
NSC	www.leav.army.mil/index.htm	National Simulation Center

NTERMS	www.Nterms.mugu.navy.mil	Navy Test &Evaluation Repository for Models & Simulation
ODCI	www.odci.gov/ic/icagen2.html	US Intelligence Community Links [ODCI]
OEA	www.nrl.navy.mil/OceanEA/	Ocean Executive Agent for M&S
OSI	www.Osi.usmc.mil	Office of Science and Innovation [Marine Corps]

SBA SIA	www.msosa.dmso.mil/sba	Simulation Base Acquisition – Special Interest Area
SCS	www.scs.org	Society for Computer Simulation
SEDRIS	www.sedris.org	Synthetic Environment Data Representation & Interchange Specification

SIA's>	www.msosa.dmso.mil/msosa-net/sia.asp	Special Interest Area [MSOSA/DMSO]
>HOBM	www.msosa.dmso.mil/hobm/	Human &Organization Behavior Modeling SIA
>IA	www.msosa.dmso.mil/ia	Impact Assessment SIA
>MSMP	www.msosa.dmso.mil/msmp	DoD M&S Master Plan Revision SIA
>MSRR	www.msosa.dmso.mil/msrr	MSRR Board of Directors &User's Conference SIA
>OOTW	www.msosa.dmso.mil/ootw/	Operations Other Than War SIA
>SBA	www.msosa.dmso.mil/sba	Simulation Based Acquisition SIA

SISO	www.Siso.sc.ucf.edu	Simulation Interoperability Standards Organization
SMART	www.Sba.iitri.org	Simulation & Modeling for Acquisition, Requirement and Training [Army SBA]
SMC/XR	www.afbmd.laafb.af.mil/org/xrm	SMC M&S Home Page [AF]

SPAWAR	www.C4iweb.spawar.navy.mil/pd13/pmw131/	Warfare Analysis, M&S Program [Navy]
SSBLaRC	www.larc.nasa.gov/fltsim/index.html	Simulation System Branch – Langley Research Center
STEP	www.acq.osd.mil/te/programs/tfr/step.htm	Simulation, Test &Evaluation Process Guidelines [OSD]
STOW	www.Stow98.spawar.navy.mil	Synthetic Theater of War – 98
STRICOM	www.stricom.army.mil	Simulation, Training, Instrumentation Command [ARMY]
TAFIM	www.library.itsi.disa.mil/tafim.html	Technical Architecture Framework for information Management [DISA]
TARDEC	www.tacom.army.mil/tardec/	Tank – Automotive research Development & Engineering Center [ARMY]
TEMS	www.nawcad.navy.mil/tems	Navy Test &Evaluation Modeling & Simulation

THUNDER	www.s#i.com/Default.htm	Homepage for THUNDER, theater – level campaign simulation [AF]
TMPO	www.tmpo.nima.mil	Terrain Modeling Project Office
TRAC	www.trac.army.mil	TRADOC Analysis Command
UK SWING	www.Siwg.dra.hmg.db/	UK Simulation Interoperability Working Group
USASMDOC	www.smdc.army.mil	U.S. Army Space & Missile Defense
USASMDC SMULATION CENTER	www.army.mil	Command U.S. Army Space &Missile Defense Command Simulation Center
USMC	www.usmc.mil	US Marine Corps Homepage
VPG	www.Vpg.tecom.army.mil	Virtual Proving Ground [ARMY]
VV & A	www.dmso.mil/dmso/projects/vva/	Verification, Validation, & Accreditation
WARSIM	www.stricom.army.mil/stricom/pm-warsim/	PM-Warfighters' Simulation [Army]

XOC	204.34.204.77/	AF Directorate of Command &Control [M&S [mgmt]

INSTITUTE OF CONTROL AND SYSTEM RESEARCH

BULGARIAN ACADEMY OF SCIENCES

Address: Acad. G. Bonchev str., Bl. 2, Sofia 1113, Bulgaria
Director: Prof. Dr. Sc. Chavdar ROUMENIN
Phone: (+359 2) 737-822
Fax: (+359 2) 737-822
E-mail: roumenin@bas.bg
roumenin@iusi.bas.bg
Web addresses: http://www.cl.bas.bg/directory/tech/icsr.html
http://members.spree.com/technology/icsr

The Institute of Control and System Research (ICSR) at the Bulgarian Academy of Sciences carries out scientific research on the problems of control, analysis and modeling of complex processes and systems.

The Institute consist of seven departments and laboratories:

1. Knowledge Based Control Systems

Field of Research: Synthesis and analysis of multilevel decentralized intelligent control systems, fuzzy control systems, neural networks and structures for control, management of research/design projects, intelligent monitoring and diagnostic systems for autonomous moving objects.

Department Head: Assoc. Prof. Valentine PENEV, Ph.D.

2. Sensor Systems and Modeling

Field of Research: Microsensors, intelligent sensor systems, microsystem technologies, non-contact automation, hardware and software for system applications in automotive, medical, environmental, security, energetics, multimedia and others. Numeric modeling of technological processes, system analysis and identification of technological processes and systems.

Department Head: Prof. Chavdar ROUMENIN, Ph.D., D.Sc.

3. Optimal Control

Field of Research: Modeling and intelligent motion control of autonomous dynamic objects via learning structures and parallel transputer nets, methods for unconditional and conditional optimization, state estimation of dynamic objects, Kalman filtering, association, integration and data fusion from sensory information systems.

Department Head: Assoc. Prof. Ognian MANOLOV, Ph.D.

4. Adaptive and Robust Control

Field of Research: Design of models for biotechnological processes: deterministic, fuzzy, and memory-based models, development of biological state observers, methods and algorithms for on-line estimation of biological variables and parameters including time delay, development of methods and algorithms for robust and adaptive control for biotechnological processes, including time delay systems and sliding mode control.

Department Head: Assoc. Prof. Trayana PATARINSKA, Ph.D.

5. Hybrid Systems and Management

Field of Research: Analysis and synthesis of complex control systems, computer-integrated control systems, distributed intelligence control systems, integrated control and management systems. Artificial and hybrid (human-computer) intelligence control systems, modeling of the human-operator, cognitive task analysis in hybrid intelligence systems, cognitive modeling for adaptive interface design, on-line task allocation in hybrid intelligence systems. Investigation and design of hardware and software for control systems, sensors and sensor systems, automation for scientific experimentation, electromagnetic compatibility in control systems, information and control systems in the National Power System, fuzzy regulators.

Department Head: Assoc. Prof. Dimcho BOYADJIEV

6. Research and Development of Technical Systems

Field of Research: Information security systems for distributed objects, radiochannel computer communications, stationary and mobile ecological monitoring systems.

Department Head: Assoc. Prof. Ivanka VIDENOVA, Ph.D.

7. Scientific Research Applications and Training – Plovdiv

Field of Research: Computer-aided design in machine construction, automated systems for information processing and control.

Department Head: Assoc. Prof. Hristo VARBANOV, Ph.D.

Actively involved in defense and security related research in modeling and simulation is the

DEPARTMENT OF "KNOWLEDGE BASED CONTROL SYSTEMS"

Sample of accomplished projects

- AntiTank Wire-Guided Missile Simulator /ATGMS/
- Tank gun simulator for T-55
- Flight simulator
- Simulator for Paraglider
- Fuzzy controlled autopilot for paraglider
- Fuzzy control for missile
- Digital Servo controller for DC motor with position and speed control on Intel 8051 (C and assembler)
- Digital servo controller for stepper motor with position and speed control on Intel 8051 (C and assembler)

Modeling of dynamic characteristics of moving platforms

For modeling dynamical characteristics of moving platforms we six or more nonlinear equations which describe linear and angular positions of rigid body in 3D space with 6 degrees of freedom. We rely on adequate dynamical models for Fagot missile, F-18, Cessna, Mig 21. Reconfigurable flight dynamics is presented in every application..

Analysis and synthesis of control loops for moving platforms

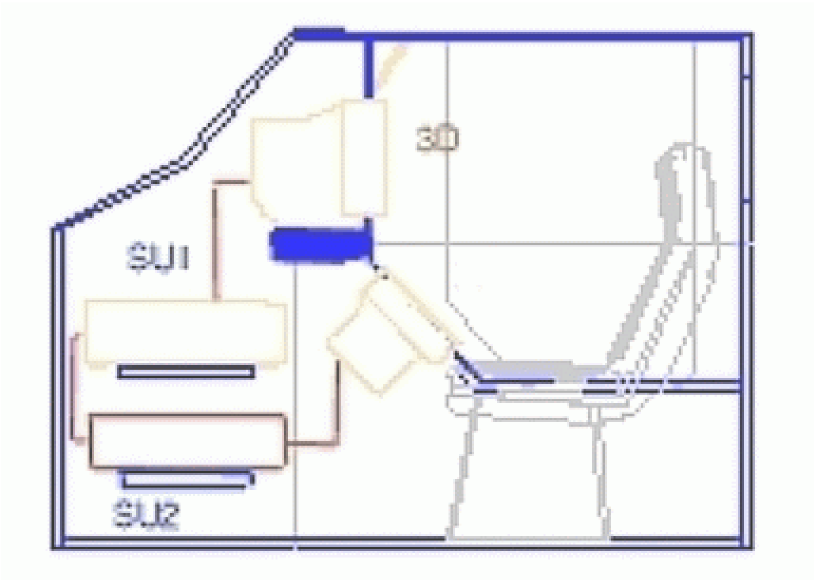
The control loops of our moving platforms were designed on the base of fuzzy control theory. Fuzzy control as an approach to nonlinear and complex control design has attracted a great deal of research interest in the past decade. The basic idea of the approach is to incorporate fuzzy IF-THEN rules into the control design, that is, fuzzy control combines two resources: input-output data and the experts' experience expressed by rules. Therefore, fuzzy control is always applied to the system, which is too complex to get the mathematical model precisely. The autopilot design for flying objects, which are highly nonlinear-coupled system, is good place to implement fuzzy control. Usually we apply fuzzy control in two general ways. First one is to use only fuzzy controller. The second one is to use fuzzy rules like tuning machine for original controllers. Of course, fuzzy control is not a tool, which can be applied everywhere. Therefore, we use fuzzy rules coupled with the transfer functions. We have developed original and elegant approaches to parametric and structural tuning of control loops.

Visualization and final design of 3D simulation software

Currently, we are developing simulation programs under Windows 95, 98 with Visual C++ 5.0 and Microsoft SDK DirectX 3 and

ActiveX. We can encompass the whole process of the simulator creation including the following:

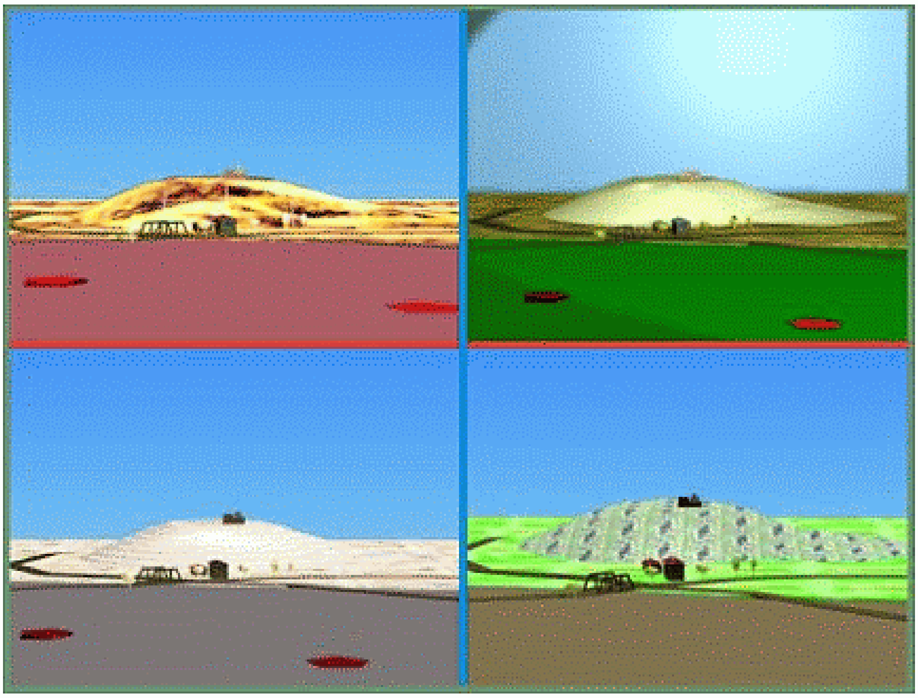
- To create the sensation of real combat conditions the computer simulation display is used. Because the system uses actual background terrain pictures, a variety of realistic environments from the desert to winter scene can be used. A number of ground features such as buildings, rivers, threes, etc., may be incorporated. The visual display simulation of weather, with particularly good representation of scud, fog are presented. Terrain is created by 3DStudio and 3D Max.
- Moving objects were designed by 3D Studio and 3D Max. The following items are presented: fighters F-4, 15, 16, 18 Mig-21, 23, 25, 27, 29, Su-27; B-1 bomber; anti-tank helicopters: AH-64, Mi-24, Mi-28; transport helicopters: K28, UH-60; armored vehicles; tanks: T-55, T-80, M1-Abrams, Merkava; missile launchers: "Squd".
- Audio equipment realistically replicates engine and weapon sounds and 3D sound effects may be implemented in software.
- Developing the software for simulators. In our application we use Direct3D, which is Microsoft's real-time, interactive 3D technology for mainstream computer users on the desktop and the Internet. Above all, Direct3D is designed for speed. Direct3D provides the API services and device independence required by developers, delivers a common driver model for hardware vendors, enables turnkey 3D solutions to be offered by PC manufacturers, and makes it easy for end-users to add high-end 3D to their systems. Because the system requires little memory, it runs well on most of the installed base of computer systems. Direct3D is a complete set of real-time 3D graphics services that delivers fast software-based rendering of the full 3D rendering pipeline (transformations, lighting, and rasterization) and transparent access to hardware acceleration. API services include an integrated high-level Retained-Mode and low-level Immediate-Mode API, and support for other systems that might use Direct3D to gain access to 3D hardware acceleration. Direct3D is fully scalable, enabling all or part of the 3D rendering pipeline to be accelerated by hardware. Direct3D exposes advanced graphics capabilities of 3D hardware accelerators, including z-buffering, antialiasing, alpha blending, mipmapping, atmospheric effects, and perspective-correct texture mapping. Tight integration with other DirectX technologies enables Direct3D to deliver such advanced features as video mapping, hardware 3D rendering in 2D overlay planes—and even sprites—providing seamless use of 2D and 3D graphics in interactive media titles. Direct3D is implemented in two distinctly different modes: Retained Mode, a high-level API in which the application retains the graphics data, and Immediate Mode, a low-level API in which the application explicitly streams the data out to an execute buffer.

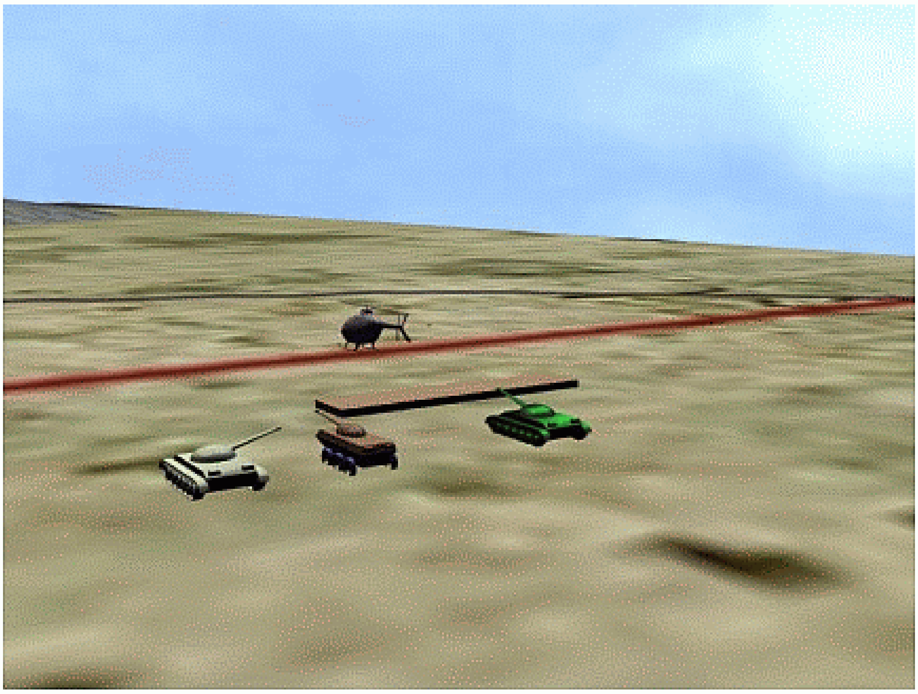


Especially for flight simulator we have a good set of flight instruments for visual and dead reckoning navigation practice. For example artificial horizon, airspeed indicator, turn indicator, inclinometer position of the throttle, altimeter, vertical speed indicator can be used depending on situations. Flight panel is realized on the single computer and 3-D environment on the second with TCP/IP. Flight simulators may be connected into group work system.

Also, we can develop the following kinds of software "Scenario maker" and "Tactical and Technical Characteristics Evaluator", which are useful tools for creating tactical scenario and evaluation of tactical and technical skills of crew.

Some background terrain pictures from the screen





Some of the moving objects

Screen view from the flight simulator

For further information:

Head of department: Assoc. Prof. Dr. Valentine PENEV

E-mail: valpenev@iusi.bas.bg

Phone: (++359 2) 70 03 37

Web addresses:

http://people.bulgaria.com/valent

http://www.cl.bas.bg/directory/tech/icsr.html

http://members.spree.com/technology/icsr