SEMANTIC CORRELATION OF BEHAVIOR FOR THE INTEROPERABILITY OF HETEROGENEOUS SIMULATIONS

by

CHRISTOPHER JAMES DEAN
B.S.E., University of Florida, 1994

THESIS

Submitted in partial fulfillment of the requirements
for the degree of
Master of Science in Computer Engineering
College of Engineering
University of Central Florida
Orlando, Florida

Spring Term
1996

ABSTRACT

A desirable goal of military simulation training is to provide large scale or joint exercises to train personnel at higher echelons. To help meet this goal, many of the lower echelon combatants must consist of computer generated forces with some of these echelons composed of units from different simulations. The object of the research described is to correlate the behaviors of entities in different simulations so that they can interoperate with one another to support simulation training. Specific source behaviors can be translated to a form in terms of general behaviors which can then be correlated to any desired specific destination simulation behavior without prior knowledge of the pairing. The correlation, however, does not result in 100% effectiveness because most simulations have different semantics and were designed for different training needs. An ontology of general behaviors and behavior parameters, a database of source behaviors written in terms of these general behaviors, and heuristic metrics are used to compare source behaviors with a database of destination behaviors.

This comparison is based upon the similarity of sub-behaviors and the behavior parameters. Source behaviors/parameters may be deemed similar based upon their sub-behaviors or sub-parameters and their relationship (more specific or more general) to destination behaviors/parameters. As an additional constraint for correlation, a conversion path from all required destination parameters to a source parameter must be found in order for the behavior to be correlated and thus executed. The length of this

conversion path often determines the similarity for behavior parameters, both source and destination.

This research has shown, through a set of experiments, that heuristic metrics, in conjunction with a corresponding behavior and parameter ontology, are sufficient for the correlation of heterogeneous simulation behavior. These metrics successfully correlated known pairings provided by experts and provided reasonable correlations for behaviors that have no corresponding destination behavior. For different simulations, these metrics serve as a foundation for more complex methods of behavior correlation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Simulation interoperability can be defined in general terms as the ability of simulations to share a common environment and work together to support a common goal. Working together may involve resolving differences in communication protocols, system behavior, system timing, etc. Sharing a common environment may involve resolving differences in system fidelity, representation, databases, environment behavior, etc. It is these differences that cause the interoperability of simulations to be a major problem in the simulation community.

In order to fully understand the interoperability problem and all its aspects, several concepts need to be discussed first. The concepts of simulation and simulation training will be defined to provide the context for the discussion. Furthermore, foundational concepts such as distributed interactive simulation and computer generated forces will be discussed to set the stage for the problems that can occur when trying to connect dissimilar simulations. Finally, the problem of interoperability, i.e. the concept of advanced distributed simulation, and its corresponding issues will be discussed. The semantic interoperability problem is but one of the many interoperability issues.

Simulation

Simulation is a technique that allows the comprehension of reality by representing it using artificial objects and acting out scenarios with them. More specifically, the modeling of reality allows the understanding of time-varying phenomena. Computer simulation is a more specific simulation discipline which involves three basic steps: the designing of the model of a physical or theoretical system, the execution of the model, and the analysis of the results [Fishwick, 1995]. There are many ways to model systems: conceptual, declarative, functional (function-based approach, variable-based approach), constraint modeling (equation-based, graph-based), spatial modeling (space-based, entity-based), and multi-modeling (a combination of different models) [Fishwick, 1995]. The system in question can be modeled under varying levels of abstraction, whichever are necessary for the needs of the problem. In cases where no one model is sufficient, the system can be modeled using a multi-model of different models at different levels of abstraction connected in a seamless fashion. Simulation models can be executed in a serial or parallel fashion and varying kinds of execution analyses can be performed such as input-output analysis, experimental design, surface response techniques, data visualization, verification, and validation [Fishwick, 1995].

Simulation plays several roles in current research. Simulation plays a role in what is termed "computational science", i.e. the visualization and simulation of large scale complex systems such as weather systems and molecular dynamics. Similarly, simulation is important in the study of chaotic and complex systems such as nonlinear systems.

Virtual reality is simulation taken to it's maximum degree, the immersion of the analyst into the simulation itself. The potential of virtual reality is enormous and yet to be fully realized. Simulation can be used to experiment with artificial life which is a topic of much debate [Fishwick, 1995]. Finally, simulation can be used in physical modeling and computer animation. Typical physical models not only share the appearance of real-world objects but obey the same physical laws.

Simulation Training

Simulation is not only used to represent reality as a means of understanding it but also for the purpose of training. Simulation training can be used with any simulation under any role. Its use is an emerging field. Currently, simulation is being used to train military personnel (infantry, tank commanders, battalion commanders, etc.), nuclear reactor operators, aircraft pilots, radar operators, oil tanker pilots, etc. It should also be noted that simulation is also being used for the design of aircraft, circuits, computers and the like. As computing power becomes more widely available, other domains that can benefit from simulation such as medicine and economics will incorporate it into their operating environment which includes training, modeling, development, planning, and design [Cohen, 1994].

In industry and the military, computer-based simulation is increasingly being used in training because it is cost effective and is able to simulate real world conditions that would otherwise be impossible to duplicate. In the military domain, large scale exercises can be simulated and used for training at all echelon levels from battalion down to the individual infantry. The worthiness of training as been shown many times, most recently during Kernel Blitz 95 [Neuberger and Shea, 1995]. The exercise showed that a Synthetic Theater Of War (STOW) created by simulation technology can provide valuable training for a variety of different roles, from actual combatants to support staff. The real-time aspect of the battlefield often makes the training of support staff difficult if not impossible. Simulation training enhances the training of support staff by making it less sensitive to the pace of the battlefield and introducing cost savings. This cost savings was suggested during Kernel Blitz 95 [Neuberger and Shea, 1995].

Distributed Interactive Simulation

To support military simulation training, the Department of Defense (DoD) has mandated the use of a framework and standard set of protocols to create a time and space coherent synthetic representation of the battlefield environment, known as Distributed Interactive Simulation (DIS). DIS is a entity-based simulation approach that allows large scale simulations to be built from independent simulator nodes which are linked via a common network protocol. Each simulator node independently simulates one or more entities and reports events over the network. A common terrain database is used to represent the shared environment. Because the simulator nodes are networked and the architecture is scaleable (within bandwidth constraints) many trainees can simultaneously participate in a training exercise and thus be effectively trained in team tactics [Petty, 1994]. DIS serves as the low level background for most military (and some non-military) simulations. It is used to support the real-time interaction of autonomous simulations, manned simulators, and equipment in live arenas. Since its main purpose is to support military simulation, the world is most often modeled as a set of combat entities (tanks, infantry fighting vehicles, infantry, aircraft, etc.) that interact with each other via events that they cause. These events are in turn perceived by other entities causing other effects and so on. Some key DIS design principles include [Institute for Simulation Training, 1994]:

- No central system that controls event scheduling.
- Autonomous simulation nodes.
- Sending nodes emit "ground truth" data, receiving nodes are responsible for perception of that ground truth, i.e. their view of the real world with environmental effects taken into account.
- To decrease network traffic, each entity uses an algorithm known as "dead reckoning" that estimates the position of itself and other entities. When the difference between the actual position and predicted position of an entity surpasses a given threshold, that entity updates the other simulators with a position update. For more on dead reckoning, see [Fishwick, 1995].
- A shooting entity determines whether a target was hit, and the target determines the damage and effect.

Additionally, the large set of critical parameters that support DIS [Humphrey, 1994] include:

- Entity performance parameters
  - Speed
  - Acceleration
  - Angle of Attack

- Perceptual limits
    - Visual
    - Audio
- Rates of fire
- Capacities (fuel, ammunition, soldiers, etc.)
- Articulated Parts
    - Enumerations (DIS characteristic constants)
    - Range of motion
    - Rates
    - Limits
- Kinds of weapons
    - Warheads
    - Fuses

The initial focus of DIS application development has been on training of large, joint, or combined forces which is lacking in traditional training. [DIS Steering Committee, 1994]. The DIS mission is defined as:

> "The primary mission of DIS is to define an infrastructure for linking simulations of various types at multiple locations to create realistic, complex, virtual 'worlds' for the simulation of highly interactive activities. This infrastructure brings together systems built for separate purposes, technologies from different eras, products from various vendors, and platforms from various services and permits them to interoperate. DIS exercises are intended to support a mixture of virtual entities (human-in-the-loop simulators), live entities (operational platforms and test and evaluation systems), and constructive entities (wargames and other automated simulations).
>
> The DIS infrastructure provides interface standards, communications architectures, management structures, fidelity indices, technical forums, and other elements necessary to transform heterogeneous simulations into unified seamless synthetic environments. These synthetic environments support design and prototyping, education and training, test and evaluation, emergency preparedness and contingency response, and readiness and warfighting." [DIS Steering Committee, 1994]

The protocol component of the DIS framework allows the simulated entities to communicate with one another and defines the various operations that can occur on the synthetic battlefield such as changes in entity state (damage, dead reckoning, etc.), firing weapons, weapons detonations, resupply, etc. [IST, 1994]. For example, the most common protocol packet sent during a DIS exercise is the entity state packet. Using it, simulated entities send location and damage information over the network which is used by other entities simulated by other simulators to generate the visual representation. Additional entity actions are communicated through collision packets, fire packets, radio communication packets, and radar/EM emissions packets. The DIS Vision defines these protocols as a:

> "set of protocols that convey messages about entities and events, via a network, among various simulation nodes that are responsible for maintaining the status of the entities in the virtual world. The characteristics of the network are not important, as long as it can convey these messages to the interested simulation nodes with reasonably low latency (100 - 300 ms) and low latency variance. Within these constraints, the systems that generate entities that appear to be adjacent in the virtual world could be separated by thousands of miles in the real world." [DIS Steering Committee, 1994]

The DIS protocol is defined over a set of protocol data units (PDUs) used for entity information, weapons fire, logistics support, collisions, simulation management, electromagnetic emissions, and radio communications. Specifically, some PDUs include:

- Entity State PDU
- Fire PDU
- Detonation PDU
- Service Request PDU

- Resupply Offer PDU
- Resupply Received PDU
- Resupply Cancel PDU
- Repair Complete PDU
- Repair Response PDU
- Collision PDU
- Create Entity PDU
- Remove Entity PDU
- Start/Resume PDU
- Stop/Freeze PDU
- Acknowledge PDU
- Message PDU
- Emission PDU
- Designator PDU
- Transmitter PDU
- Signal PDU
- Receiver PDU

DIS is meant to be the canonical paradigm for distributed interactive simulation. Unfortunately, this is not entirely true as will be shown later. There are many issues involved in the support of DIS [Cohen, 1994] but these are not of concern here. What is of concern are the issues involving the use of Computer Generated Forces (CGF) and Semi-Automated Forces (SAF) for training in a DIS environment.

Computer Generated Forces

DIS simulations usually include special simulation applications known as CGF or SAF nodes. Because of the human resources needed to train upper and/or lower echelon personnel in large combat situations, CGFs are needed to provide a more robust training environment without additional manpower. CGF systems initially came into being as a result of the need to provide threat vehicles or supplementary friendly forces to train personnel on simulators. CGF is able to provide realistic complementary forces and enemy forces. For example, at the Joint Readiness Training Center, CGF is used to represent one of three battalions and the corresponding OPFOR (opposing force) to provide realistic training of command and control for the regimental and battalion commanders [Jones, 1993]. In addition, the CGF can be used to represent any special force elements needed, joint force elements, and even coalition forces. Since 1990, CGF systems have been augmented to act not only as a simulator of threat and friendly vehicles but to act as a virtual experimentation environment. CGF permits experimentation with new doctrine and operational plans over diversified conflicts and experimentation with new equipment (this is known as a Battle Lab within the military) without the expenditure of the considerable time and money necessary to conduct a field exercise with geographically dispersed assets [Jones, 1993]. As an example, CGF was used in a scenario that involved using the Army Tactical Missile System (ATACMS) to destroy time-critical, high-priority targets such as air defense sites [Jones, 1993]. CGF was used to evaluate the length of time from target acquisition through the decision process and weapon-on-target in this time-critical operation. This usage has resulted in the various kinds of simulations that have been and are continuing to be developed to meet the diversified needs of the military. These additional features include the ability to support a virtual battlefield composed of not only vehicles and aircraft but three dimensional terrain and battlefield environment. In addition, these virtual simulators have been interfaced with live personnel and sensor simulators such as J-STARS to provide command and control decisions. The CGF was used to provide J-STARS information (which was evaluated for usefulness), providing trainees with the opportunity to work with new systems before they are fielded and obtain otherwise hard to obtain data on behavior and response times [Jones, 1993]. An example from Jones [1993] includes the analysis of the timeliness of AWACS operators associated with controlling interceptors. The analysis provided important data on the survivability of tactical and strategic low observable vehicles. With the virtual CGF systems, new attack capabilities, force structures, tactics, equipment, and Tactics, Techniques, and Procedures (TTPs) can be demonstrated and analyzed. Typical examples of these experiments include the Advanced Warfighting Demonstrations (AWDs) and Advanced Technology Demonstrations (ATDs) [Pickett and Petty, 1995]. In addition to training, CGFs serve as a device for operations planning and mission rehearsal. Portions of a mission can be practiced or rehearsed as part of on-going training and readiness of a component of a force. This can identify problem areas or weaknesses of the planned mission. Military commanders can also use the results of CGF to be gain experience and exposure to the various eventualities that may occur during the mission.

Currently, a military simulation may be virtual, constructive, or live. Each varies in the training resolution, timing mechanisms, and user interactions. Virtual simulations exhibit a high resolution representation of the battlefield, often simulating individual entities that make up larger units (tanks, dismounted infantry). They are typically real time and interact with the user in an asynchronous, time-driven manner. Because of their resolution, virtual simulations only provide a limited set of combat events. Constructive simulations are lower resolution simulations designed to train upper level tasks such as battalion command and logistics (medical and supply, for example). Groups of entities are represented in an aggregate manner with individual entity actions and results simulated using force probability functions such as Lanchester equations [Taylor, 1983]. Lanchester equations are simple equations used to measure combat attrition, i.e. to decide which force is affected and how much strength is lost on an aggregate, not unit, level. Constructive simulations sacrifice entity and event detail for the breadth of operations that can be performed. Typically, constructive simulations run in faster than real time but can be adjusted to any time frame. They interact with the user in a synchronous time step fashion based upon events. Finally, live simulations can exhibit properties of both. High resolution is used for device specific actions and lower resolution for auxiliary actions. Live simulations may be made to interact with virtual and constructive simulations but since there is no computer simulated aspect, they will not be addressed further. Both constructive and virtual simulations make use of CGFs and SAFs to enhance training. CGFs are often used to represent the actions of opposing forces (OPFOR) as well as representing additional units (such as platoons) for friendly forces (BLUFOR). The only difference between constructive and virtual as far as CGFs are concerned is the resolution of representation and resolution of behavior. For an overview of the various constructive and virtual simulations see [Sandmeyer and Dymond, 1995]. For constructive simulations, the CGF component is concerned with the automation of command and control decision making. Command and control is the process of analyzing the situation and issuing orders to subordinate units. For virtual simulations, the CGF is more concerned with reactive behaviors such as reacting to an enemy attack. SAFs are similar to CGFs in that SAF units contain CGFs at the lower echelons with a trainee at the higher echelons such as the company or battalion commander. Since the trainee controls the lower echelons via orders, they are termed semi-automated. The unmanned units still are responsible for their reactive and primitive behaviors (move, shoot, etc.). SAF arises from the need to have experienced soldiers in the simulation to interject behaviors into the system that are difficult to simulate automatically with present technology. Another advantage for CGF, aside from the human resource issue, is that using CGF for lower echelon units allows the pace of the exercise events to be tailored to the handling capacity of the echelon being trained without concern that the planning phases will generate delays in the action for lower echelon units [O'Byrne, 1993].

CGF systems are characterized by a set of standard objectives/benefits [Jones, 1993; Weaver, 1993; Picket and Petty, 1995]:

- CGF systems must support training, advanced technology demonstrations, and analysis which includes support for man-in-the-loop simulators at any echelon level, live interfaces at any echelon level, real-time or faster than real-time processing, and constructive-virtual interfaces.
- CGF systems must provide a realistic operating environment and varying scenarios.
- CGF systems should provide analytical summary information on the exercise.
- CGF systems must be DIS compatible and all that being DIS compliant implies.

- CGF systems must represent training from the individual vehicle/infantry up to the corps level.
- CGF systems should be able to interface with other service simulations that would be required in a joint exercise.
- CGF behaviors should be written in a modular and low coupled fashion to support verification and validation.
- CGF systems should be able to operate in any simulated environment.
- CGF systems can provide any fraction of an exercise.
- CGF systems should keep the number of required manned operators to a minimum. This is the focus of CGF behavior research.
- CGF behavior should be indistinguishable from the behavior of the human participants on the battlefield. This is also an important issue when considering behavior generation.

Implicit in the term CGF is the expectation of some form of intelligence and representation of decision making. Since CGF systems are used to represent OPFOR and supplementary BLUFOR forces, they must exhibit a degree of realism that allows trainees to receive positive training benefits [Petty, 1994]. CGF OPFOR units must fight like an enemy would and BLUFOR CGF units must cooperate with the trainee's force(s), i.e. they must react to the simulated situation and perform intelligent and doctrinally correct actions. The Institute for Simulation Training has researched the use of the Turing Test [Turing, 1950] as a criterion for CGF [Petty, 1994]. The Turing Test has many variants [Petty, 1994] but the most widely known formulation of the test is for an interrogator to determine, using a series of questions, if a respondent is a human or computer system. There is much controversy associated with the use of test as a criterion for intelligence, but for CGF the basic question is "Can observers of simulated entities in the battlefield reliably determine whether any given entity is controlled by humans or a CGF system?" [Petty, 1994]. The Turing Test makes no restrictions on how the behavior is generated and as will be shown later, can vary greatly in implementation.

Passing the CGF Turing Test is easier than passing the original Turing Test since the "observer" is a trainee, not an expert on unit tactics. Also, combat operations are always conducted within the context of national and service doctrine. The doctrine is expressed at unit levels in the form of training, equipment and planning routines, operational techniques, and functional agencies that manage specific aspects of combat operations. Any commander's decisions/plans, and thus any CGF decisions/plans, are constrained within this context. Thus, the set of possible interactions that could occur is limited to only that doctrinal set allowed by the simulation. The actions of the trainees become the questions and the CGF actions become the responses. Also, the trainee's ability to observe other units is severely constrained by the fact that the trainees are only allowed to observe the portion of the battlefield that is visible from their location and with the visual equipment allowed by their simulated vehicle. Aggravating this situation is the fact that, in the case of OPFOR units, the units are trying to remain concealed. This restricted ability to observe the battlefield limits the ability of trainees to determine whether the units are human or computer controlled. Finally, [Petty, 1994] suggests that the trainee is probably more concerned with some activity that is pertinent to the current mission such as destroying opponents quickly as possible (missions usually have limited time) rather than observing the opponent's behavior for signs of artificiality. See Petty [1994] for a description of some of the CGF Turing Test experiments. The experiments were designed to measure whether the CGF Turing Test is sufficient (a system passing it will surely produce positive training), necessary (system must pass it in order to be able to produce positive training benefits), or irrelevant (passing does not matter). The author concludes that the CGF Turing Test is irrelevant but does provide a useful heuristic to replace it. The experiments demonstrate that it is possible to produce positive training benefits from a system that does not pass the test and to not receive training benefits from systems that do. Passing the test only serves as evidence to the quality of the behaviors present in the system.

Thus far the discussion has centered primarily on CGF for virtual simulations. CGF also applies to constructive simulations at higher echelons, especially the joint task force (JTF) level. At this level, there are a number of agencies that can benefit from simulation training including the following [O'Byrne, 1993]:

- TACC Tactical Air Command Center (USMC)
- TACC Tactical Air Control Center (USAF)
- TACC Tactical Air Coordination Center (USN)
- TADC Tactical Air Direction Center
- TACLOG Tactical Logistics
- TAOC Tactical Air Operations Center
- DASC Direct Air Support Center
- FSCC Fire Support Coordination Center
- COC Combat Operations Center
- MTMC Military Transportations Movement Center
- DLA Defense Logistics Agency
- DIA Defense Intelligence Agency
- CRC Control and Reporting Center

- HDC Helicopter Direction Center
- TOC Tactical Operations Center

Each agency has resources assigned for management and logistics functions. These assets are balanced against the planned operations of the JTF commander. This relationship and the agency authority are specified again by doctrine. The important aspect of these agencies is that they require 15-75 specially qualified personnel to operate. CGF can play an important, and in some cases more difficult, role in training these agency personnel by replacing agency personnel with computer generated equivalents. The CGF can also train command post personnel by sending reports (situation reports, spot reports), starting at any echelon level, up the command hierarchy.

As mentioned previously, CGF serves a role in testing new equipment. [Jones, 1993] contends that this system acquisition and development is actually its greatest benefit. CGF can be used to support weapon requirements development (Cost and Operational Effectiveness Assessment -- COEA), early operational assessments (EOA), and development evaluation. The principal advantage is the ability to use a common methodology throughout the system acquisition and development process. An analyst can then focus on the results, rather than on the assumptions of the different system methodologies behind those results, because he/she better understands the system. Field test data allows the system to evolve and can serve to validate or adjust the CGF and its associated data bases, giving the analyst more confidence in the system and its results [Jones, 1993]. The CGF provides an operational context for requirements assessment and provides important data that is frequently lacking when assessments are conducted. Using CGF, a virtual battlefield can be created with soldiers fielding experimental equipment against a realistic threat and thus deployment doctrine can be adjusted before any actual weapon construction is begun. The CGF can provide real time kill assessments with simulated test participants.

CGF also plays a role in pre-test analysis/test planning and post-test analysis. For pre-test analysis, CGF can identify critical factors to measure, scenario sensitivities, establish field test scenarios and event timelines, and predict outcomes, providing the "where" and "when" for the new equipment to be tested. In post-test analysis, CGF can be used to fill in missing information, extrapolate information, and translate those results to different locales. CGF can help determine the operational effectiveness and suitability of the new system under different robust, operationally realistic scenarios.

Advanced Distributed Simulation and Interoperability

As previously shown, CGF systems can span a wide range of fidelity and focus on many different aspects of training. Because no existing CGF system satisfies all the requirements for all users, CGF systems must interoperate with one another. All the various missions established as the vision for DIS (including civilian domains such as aviation command and control, disaster relief, distributed simulation games, and team training efforts) create specific challenges to simulation training. Attempts to meet these requirements with the flexibility and fidelity required leads to the interoperability of various CGFs, constructive and virtual. This interoperability is defined as Advanced

Distributed Simulation (ADS), and implies an ongoing evolution of simulation. The ADS environment, which may be synthetic or virtual, represents real world phenomena for the purpose of training, testing of developing systems, analysis, doctrine development, etc. It is the logical extension of DIS for the purposes of multiple heterogeneous simulations interoperating in a common environment. Also, over time, simulations will evolve and may require additional information and/or features not currently anticipated and defined, thus requiring interoperability adjustments. Interoperability can be simply defined as the

> "set of explicit expectations (rules) and implicit expectations (assumptions) which are made by users in a simulation exercise " [Riecken and O'Brien, 1994].

Expressed differently, interoperability can be defined as a measure of consistency between representations of the simulated environment. By one definition, interoperability has been achieved if the perception of the virtual space is sufficiently similar when viewed from different simulations [Altman et al., 1994]. By another definition, if the simulated outcomes match the desired training outcomes, then interoperability has been satisfied [Moskal et al., 1994]. Regardless of the definition, interoperability problems must be dealt with.

More specifically, ADS interoperability can be defined at two primary levels, the application and core level. The core level includes interoperability between network interfaces, software architectures, languages, and data representation. Development of standard interoperable software modules is an issue at this level. The application level is concerned with the interoperability between simulations and/or simulation components.

At this level, interoperable simulations can be defined as simulations using compatible protocols (valid in communicating what is being done), simulations using compatible algorithms (valid in determining how operations are being done), and in some cases simulations using compatible design requirements (valid in why operations are being done). [Smith, 1995] goes on to say that due to the limited understanding of complex processes, limitations of modeling fidelity, and increasing lack of process determinism, systems may be valid when run alone (meet the communication, algorithm, and design criteria) but invalid when combined with other simulations. The DIS protocol was an attempt to alleviate this problem. Unfortunately, the DIS standard does not provide for all types of interoperability. DIS was developed under the myth that the exchange of data would guarantee interoperability [Altman et al., 1994]. As mentioned previously, DIS does provide standards for interface definition, communication, environment representation, management, security, field instrumentation, and performance measurement. However, it does not specify entity representation standards, behavior standards, synchronization standards, or spatial coherence (correlation of terrain, resolution correlation and environment correlation such as ambient illumination, buildings, weather, etc.) standards and database standards. DIS can deal with limited forms of interoperability such as sensing interoperability, direct interactive interoperability, indirect interactive interoperability, associative interoperability, communications interoperability, and simulation management interoperability [Rush and Whitely, 1994]. Sensing interoperability is the ability for a battlefield element to sense another (fair fight) either visually, thermally, with radar, etc. While DIS supports this, it does however have a problem went interoperating different kinds of simulations as will be discussed later. Direct interactive interoperability is the ability for a battlefield element to physically interact with another such as moving over terrain or collisions. Indirect interactive interoperability is similar, only an indirect method of contact such as shell fire is involved. Associative interoperability is the ability for battlefield elements to act as though they were connected to another element such as vehicles moving in formation. Again, as will be shown later, if the battlefield elements are controlled by different simulations, DIS does not support complete associative interoperability. Finally, communication interoperability is the ability for elements to communicate with one another and simulation management interoperability is the ability to examine or control the parameters of battlefield elements.

CHAPTER 2

BACKGROUND

Behavior Representation for CGF

Behavior for CGFs have been loosely grouped into two categories: reactive and planning. Planning behavior is a traditional AI research area that offers the advantage that it is suited to general-purpose problem solving, thus allowing decisions to be made in unfamiliar situations or with unanticipated goals. Reactive behavior has an advantage over traditional planning in that CGFs using reactive behavior can react more quickly to a changing environment and thus operate more robustly in the dynamic, sometimes unpredictable, world of military simulation. As will be shown, both are used for CGF systems. It is also becoming clear that hybrid approaches using both types will be needed as the future of CGF increases in complexity. The various levels of complexity can be seen by analyzing the various kinds of behavior a CGF can be expected to perform.

CGF uses various forms of declarative modeling to generate behavior. CGF behavior can be defined on three levels: the individual level, crew level, and unit level [Pratt et al., 1994]. Individual level behaviors are characterized by decisions that are updated continuously by analysis of a priori alternatives, usually generated towards a specific goal. Typical individual behaviors include firing a specific weapon, scanning an area, or seeking cover. The crew level is characterized by collaborative behavior. A crew commander coordinates the behavior of his soldiers to accomplish the assigned mission. The roles of each crew member vary from steering a vehicle, rotating a turret, loading a weapon, firing a weapon, etc. The crew commander has to consider more decision factors and alternatives than an ordinary individual. The unit level is characterized by the coordination of behavior. This becomes more difficult at higher levels of the military command hierarchy. The primary three functions of unit-level decision making are Command and Control (C2), route planning, and target engagement, all of which must be considered in order to provide realistic CGF. Command and control is characterized by tactical decision making, task assignment, target assessment, target assignment, fire control, and communication. Route planning is characterized by goal directed reasoning, terrain analysis, threat analysis, and vehicle/unit movement. Target engagement is simply characterized by operational decisions such as terrain assessment, sector scanning, target acquisition, weapons selection and firing. Unit level behavior is further characterized by many alternatives and difficult situational awareness above and beyond that needed for crew and individual behaviors. The lower levels focus on just the execution of a task but the unit level must also consider the selection of tasks, assignment of tasks, and coordination of those tasks. Typical tasks include movement in formations, assaults, occupying positions, etc.

Command and control behaviors consist of the planning and coordination tasks necessary for a combat unit to achieve its goals. Actions include directing the movement