REQUIREMENTS FOR INTELLIGENT AIRCRAFT ENTITIES IN DISTRIBUTED ENVIRONMENTS

Lt. Colonel Martin R. Stytz, Ph.D., Major Sheila B. Banks, Ph.D., Eugene Santos, Ph.D.
Virtual Environments, 3D Medical Imaging, and Computer Graphics Laboratory
Artificial Intelligence Laboratory
Department of Electrical and Computer Engineering
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433
mstytz@afit.af.mil, sbanks@afit.af.mil, esantos@afit.af.mil

ABSTRACT

For computer generated forces to be useful in training environments, they must exhibit a broad range of skill levels, competency at their assigned missions, and comply with current doctrine. Because of the rapid rate of change in Distributed Interactive Simulation and the expanding set of performance objectives for any computer generated force, the system must also be modifiable at reasonable cost and incorporate mechanisms for learning. The requirements pose an intricate set of challenges because the system must satisfy reasoning and fidelity requirements as well as performance requirements. To address these circumstances, we developed a set of general requirements for aircraft computer generated forces and used them to guide our specification of a generalized architecture for aircraft computer generated forces. In this paper, we present a component-wise decomposition of the system and describe the structure of the major components of the computer generated force decision mechanism. We illustrate the application of this architecture by presenting its application to the design of an aircraft computer generated force, the Automated Wingman.

ABOUT THE AUTHORS

Martin R. Stytz is an active duty Lieutenant Colonel in the U.S. Air Force serving as an Associate Professor of Computer Science and Engineering at the Air Force Institute of Technology. He received a Bachelor of Science degree from the U.S. Air Force Academy in 1975, a Master of Arts degree from Central Missouri State University in 1979, a Master of Science degree from the University of Michigan in 1983, and his Ph.D. in Computer Science and Engineering from the University of Michigan in 1989. He is a member of the ACM, SIGGRAPH, SIGCHI, the IEEE, the IEEE Computer Society, the Engineering in Medicine and Biology Society, the IMAGE Society, the Virtual Reality Society, and the Society for Computer Simulation. His research interests include virtual environments, distributed interactive simulation, modeling and simulation, user-interface design, software architecture, and computer generated forces.

Sheila B. Banks is an active duty Major in the U.S. Air Force serving as an Assistant Professor of Computer Engineering at the Air Force Institute of Technology, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH. She received a Bachelor of Science in Geology, Magna Cum Laude from University of Miami, Coral Gables, FL in 1984 and a Bachelor of Science in Electrical Engineering, Summa Cum Laude from North Carolina State University, Raleigh, NC in 1986. Also from North Carolina State University, Raleigh, NC, she received a Master of Science in Electrical and Computer Engineering in 1987 and her Doctor of Philosophy in Computer Engineering (Artificial Intelligence) from Clemson University, Clemson, SC in 1995. Her research interests include artificial intelligence, intelligent computer generated forces, associate systems, distributed virtual environments, intelligent human computer interaction, and man-machine interfaces.

Eugene Santos Jr. is an assistant professor of computer science at the Air Force Institute of Technology. He received his B.S. in Mathematics and Computer Science (1985) and M.S. in Mathematics -- Numerical Analysis (1986) from Youngstown State University (1985) and subsequently completed a Sc.M. (1987) and Ph.D. in Computer Science -- Artificial Intelligence (1992) from Brown University. His research interests include automated reasoning, neural networks, natural language understanding, expert systems, machine learning, operations research, probabilistic reasoning, robotic planning, temporal reasoning, combinatorial optimization, numerical analysis and parallel processing. Member, IEEE, ACM, Sigma Xi and AAAI.

REQUIREMENTS FOR INTELLIGENT AIRCRAFT ENTITIES IN DISTRIBUTED ENVIRONMENTS

Lt. Colonel Martin R. Stytz, Ph.D., Major Sheila B. Banks, Ph.D., Gene Santos, Ph.D.
Virtual Environments, 3D Medical Imaging, and Computer Graphics Laboratory
Artificial Intelligence Laboratory
Department of Electrical and Computer Engineering
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433
mstytz@afit.af.mil, sbanks@afit.af.mil, esantos@afit.af.mil

INTRODUCTION

Practical development of intelligent entities for distributed interactive simulations and advanced distributed simulations requires addressing a wide variety of issues related to knowledge engineering, information structure, decision making, scalability, and system software architecture. To date, the majority of systems have addressed these problems only in passing because of the need to meet program schedules. Additionally, the need for realism in the simulation environment as well as real-time performance considerations, have forced many systems to adopt ad hoc, non-scaleable solutions to these problems to achieve real-time performance.

The requirements for intelligent computer generated forces (CGFs) stem from the need to reduce operational costs while providing a realistic training and analysis environment that operates in real-time. Allow us to briefly review and summarize these requirements here, we will return to them in greater detail later in the paper. CGF requirements address the need for modifiability, high fidelity representations, adaptable decision mechanisms and behaviors, and automated incorporation of past reasoning into the decision process. The CGF must also exhibit multiple skill levels for classes of entities, graceful degradation of reasoning capability under system stress, an easily expandable modular knowledge structure, and adaptive mission planning. For example, the capability of multiple skill levels would provide pilot behaviors for aircraft entities at different skill levels. This would provide rookie, expert, and ace levels of pilots within the same battlespace, which should make for better training and analysis. Additionally, the entity should have a complete set of behaviors for the type(s) of missions it must perform, but not all behaviors need to be crafted to the same level of fidelity and quality. Merely considering individual entity behaviors is not sufficient. Issues related to complex inter-entity behavioral interactions, such as the need to maintain formation, must also be considered. Because CGF performance must continually adapt to new requirements, the system should be customizable by users in the field and be adaptable to new weapon systems, vehicles, and tactics. The CGF should be able to respond to unforeseen circumstances with an acceptable response and be able to deal with uncertain information. Because of the need for increased complexity in the virtual battlespace, the CGF should exhibit complex, realistic behavior patterns within the battlespace during the course of its mission and be able to adaptively change mission parameters during the course of a mission in response to events. Finally, these capabilities should be embedded in an extensible, explicable, layered software architecture that has well defined locations for reasoning and knowledge storage.

In this paper, we will review necessary background material driving aircraft CGF requirements and background concerning a methodology of addressing these requirements. The foundations and motivations for aircraft CGF requirements, their implications, and their importance are then discussed. A means for successfully addressing these requirements is presented. The paper closes with a summary of our approach and our assessment of needs for future development.

BACKGROUND

This section discusses the topics of Distributed Interactive Simulation (DIS), current aircraft CGF background and projects, and fuzzy logic. These topics are necessary to understand the subsequently derived requirements for aircraft CGFs and the solution methodology that addresses these requirements.

Distributed Interactive Simulation

The most widespread use of network technology for distributed virtual environments (DVEs) relies upon the current DIS suite of standards (IEEE Standard 1278-1993). DIS was designed to link distributed, autonomous hosts into a real-time distributed virtual environment via a network for exchanging the data that describe events and activities. DIS takes the concept of environmental distribution to its extreme; there is no central computer, event scheduler, clock, or conflict arbitration system. Each entity is responsible for processing the information that it receives and for insuring that all remote hosts have current and accurate information about its state. Stytz presents additional information concerning DIS and DVEs (Stytz, 1996a).

Current CGF Background and Projects

Computer generated forces that exhibit human-like behaviors are critical to achieving large-scale distributed virtual environments. Their presence is important because they permit a complex virtual environment with a large number of actors to be activated without the expense of involving large numbers of humans. The challenges for a CGF lie in computing human-like behaviors and reactions to a complex dynamic environment in real-time, or at least at a human-scale rate of time. However, the challenge is eased somewhat because there is no need to replicate the human decision process, instead only the observable aspects of human decision making must be mimicked. However, the CGF behavior must be realistic and accurate enough so that other CGFs and human participants respond to it as if it were a human-controlled actor. Advances in artificial intelligence are important to the development of realistic CGFs. The capability to construct large, complex reasoning systems and the development of large knowledge bases for use by the decision machinery combine to enable the implementation of CGFs of acceptable fidelity.

For our purposes, the major components of an aircraft CGF include the following: vehicle dynamics, behavior modeling, artificial intelligence, and software architecture. Vehicle dynamics are important in CGFs because the actor should move through the virtual environment accurately whether it is human or computer-controlled. The vehicle dynamics for computer-controlled actors should not allow a human to identify it as a CGF by virtue of either exceptionally good or poor motion or display of dynamic behavior that would identify it as being computer controlled.

Human behavior modeling addresses the task of making the behavior and reactions of a CGF seem realistic by developing models that yield a reasonable analog of the output of the human decision-making process. The human behavior modeling subcomponent of CGFs might normally be considered as part of the artificial intelligence subcomponent. However, separating it from artificial intelligence serves to highlight its importance and the need to model behavior independently from the approach used to act upon the behavior model. In addition, the human behavior modeling component is the focus for certifying the accuracy of the performance of the CGF. Human behavior modeling requires acquiring domain-specific knowledge about the human mental models and information brought to the decision-making process and the key factors in the process. For military virtual environment purposes, this involves incorporating doctrine, tactics, knowledge models, information, and training into the behavior of the CGF, making this component relatively static but nevertheless complex.

The area of artificial intelligence addresses the problems associated with processing the situation considering constraints like plans, mission, activities by other actors, domain knowledge, and the capabilities of the vehicle that the CGF must control. The artificial intelligence component insures that the CGF pursues its goals, responds in a proper, human-like manner based upon its knowledge base, keeps the performance of the CGF within human and sensor limits, develops plans based upon its knowledge base, and manages other tasks. Fielded systems, like TAC-AIR SOAR (Tambe, 1995) and ModSAF (Calder, 1993) address these problems at different levels of fidelity. TAC-AIR SOAR is the most successful of the current aircraft CGFs, which builds upon the Soar architecture (Laird, 1987) for general intelligence and reasoning. However, TAC-AIR SOAR does not handle uncertainty in its decision making process.

The vehicle dynamics, behavior modeling, and artificial intelligence system components are brought together within the CGF software architecture component. A flexible CGF software architecture ensures that current CGF development efforts are extensible to future CGF requirements. The ability to modify the implemented CGF to include additional behavioral requirements is directly attributable to the software architecture's flexibility. In addition to the incorporation of new behavioral requirements, a flexible CGF software architecture allows the system to effectively address computational efficiency, incorporation of additional knowledge engineering efforts, and scaleable performance.

Fuzzy Logic

Within a fuzzy-logic system of reasoning, an assertion may have a degree of both truth and falseness. While this may seem confusing at first, it is a common way to represent situations. For example, consider a piece of teal matte board and the assertions "the board is blue" and "the board is green." Depending upon the shading, we may say that the assertion that the board is blue is true to degree 0.6 (out of 1) and the assertion that the board is green is true to degree 0.4. We now have two assertions, both true to a degree. We can reason with these assertions by factoring in the degree of truth of each assertion to arrive at a conclusion that considers all of the available information.

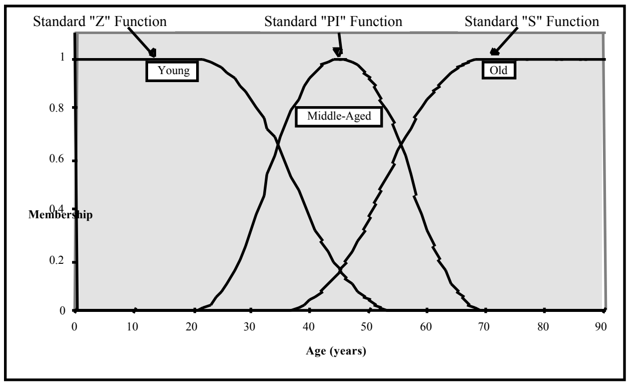

Figure 1. Specification of the AGE Linguistic Variable

The concept within fuzzy logic most applicable to the AW project, the aircraft CGF development effort described later in this paper, is that of a *linguistic* *variable*. A linguistic variable, such as temperature, describes a quantity or an idea that is best represented by fuzzy sets, called *term sets*. A more powerful technique is to *fuzzify* a crisp value and determine the term set(s) to which the crisp value belongs, allowing the linguistic variable to be valuated as the union of its fuzzy sets. The linguistic variable then takes on the value of all the term sets that apply, not the crisp value itself (Zadeh, 1975 and Schwartz, 1991). An example term set is age. The concept of age is often described in terms of young, middle-aged, and old. Figure 1 shows one possible design for the linguistic variable Age. In this example, a 15 year-old person is "young" while a 75 year-old person is "old." A 45 year-old person could be considered "young" or "old"' to a very small degree but is primarily called "middle-aged."

COMPUTER GENERATED FORCES REQUIREMENTS AND THEIR IMPLICATIONS

Even though we rely upon a rapid experimental and evaluation prototyping approach to system development (described in Stytz, 1996b), a set of baseline requirements are necessary to provide a foundation for system development. This section broadly describes these requirements. In the next section we describe how we approach these requirements.

CGF Requirements

The requirements for a CGF can be broken down into several categories. These range from the software architecture that implements the CGF to the knowledge base that is used by the CGF to support its decision making. The genesis for these requirements is the need to support a wide variety of training scenarios at the lowest possible unit cost while simultaneously achieving a credible representation of the behavior of the modeled entity. We open this section with a discussion of each requirement: 1) modifiability, 2) high fidelity representations, 3) adaptable decision mechanisms and behaviors, and 4) automated incorporation of past reasoning into the decision process. Subsequent subsections provide additional background on each requirement and assess the development and performance implications of each requirement.

Modifiability. In our view, modifiability is the ability to enhance existing CGF capabilities and includes rapid expandability of the domain-specific knowledge base and a flexible software architecture. The requirement for knowledge base expandability focuses on the need for the CGF to incorporate new strategies, tactics and maneuvers to reflect current ally and opponent concepts. If this requirement is ignored, then the CGF stagnates and large expense will be incurred to maintain the usefulness of the system for training. A flexible software architecture likewise insures that the CGF can be readily adapted to meet new performance, interface, and communication protocol requirements. The architecture should be able to support system development as well as the fielded system. While the software architecture should not change often once the system is fielded, there will continue to be system upgrades and the software architecture should be able to incorporate these changes with limited repercussions on the rest of the system.

High Fidelity Representations. High fidelity representations in the CGF are achieved by enabling CGF operation using an accurate 1) world representation, 2) dynamics for vehicle motion, 3) sensor and weapons models, and 4) model of human behavior. The world representation is based upon surface representations using primitives within a hierarchically organized representation of the terrain data. However, to simply have a high fidelity representation within the CGF is not sufficient. The CGF must also interact with other entities within the distributed training environment; therefore, the representation must also permit a high fidelity data interchange capability. In short, since CGFs do not operate in isolation, their world representation must have a high fidelity counterpart for manned systems as well as for other CGF systems. The issue of implementing correct dynamics for vehicle motion is another aspect of achieving a high fidelity representation. The vehicle dynamics concerns issues associated with insuring that the vehicle only moves according to its capabilities and does not achieve a level of performance that is unrealistic given the terrain, weather, and atmospheric conditions. Likewise, the weapons and sensor models must properly report the same field of view and range as its real-world counterpart. This level of fidelity must range from the modeling of the eyesight of the operator of the CGF to the RADAR and Infrared sensing systems of the CGF.

The most difficult challenge lies in modeling human behavior. To be a useful model for training purposes, we believe that the model must incorporate the following characteristics: 1) correctly computed outputs of the human decision making process at human-scale rate of time, 2) unpredictability, and 3) certifiability. The first component, correct outputs modeling, forces the CGF to react at a human-scale rate of time and requires that the decisions appear to be made by a human. That is, random, disjointed sequences of decisions, decisions that seem to be made irregardless of the world state, and decisions that seem to be made in total disregard for the success of the mission or preservation of the CGF should not occur. Additionally, for the entity to operate believably in the virtual environment, it must produce the appropriate behaviors/operations in a human-scale rate of time or at least within the time bounds of the distributed training environment.

The second component, unpredictability, addresses the need for CGFs to behave in a manner such that human opponents can not detect patterns in the behavior and then use these patterns to defeat the CGF. The CGF must not exhibit a pattern in its decision making, thus, forcing its human or CGF opponent to rely on its training to defeat it. Predictable patterns in CGF behaviors lead to negative training because the human operators will learn to defeat the automated opponent by exploiting these patterns of behavior rather than exploiting their doctrinal/tactical weaknesses or weapon system vulnerabilities.

The third component of human behavior modeling is that the behavior be certifiable. To be certified, a CGF software system must be able to be measured against and compared to the exhibited behaviors of a human in a comparable situation. A certified CGF system, therefore, is one that has exhibited credible behavior within a set of test scenarios. It does not mean that the system is provably correct or that its responses are credible in all situations. The need for certifiability drove our decision to model behaviors rather than to model the human decision making process because behaviors may be observed and measured.

Adaptable Decision Mechanisms. Adaptable decision mechanisms allow the CGF to exhibit a degree of flexibility in dealing with situations that occur in the virtual environment. The decision mechanisms must adapt to the amount of information that is available and to requirements for different levels of CGF performance in the battlespace. Adaptable decision mechanisms permit the system to maintain robust, credible behavior for the CGF at run-time under a variety of external circumstances and at different levels of operator skills. The first sub-requirement, robust, credible behavior, is necessary so that the CGF continues to act and react even when confronted by conflicting or incomplete information and when under system stress. If the CGF does not exhibit robust behavior, then the CGF will fail or have a scripted pattern of behavior. The second sub-requirement addresses the necessity to provide multiple levels of operator skills for a class of CGFs. Multiple skill levels allow the training to be tailored to the skills of the human participants and provide a more realistic training situation because the opponents and allies exhibit a variety of capabilities to train against; therefore, the training environment is more realistic.

Automated Incorporation Of Experiences Into The Decision Process. The requirements outlined above are difficult to satisfy because they must be addressed using large software systems that have extensive knowledge bases. The interplay between the knowledge bases and decision mechanisms further complicates the task of implementing a system to meet these requirements. Large systems are surpassing the ability of a single person to understand the interrelationships between the knowledge base and decision mechanisms for complicated systems. Therefore, CGFs must incorporate a learning mechanism into their architecture so that, by virtue of participating in training environments, they will improve their decision making capability.

Implications Of These Requirements

The requirements for modifiability, high fidelity representations, adaptable decision mechanisms and behaviors, and automated incorporation of past reasoning into the decision process have implications for system complexity, real-time performance, knowledge engineering, and scalability. Table 1 summarizes these implications. We discuss the implications in detail below.

The requirement to achieve a *modifiable* system indicates that the system should be structured so that components are isolated from each other and so that there is loose coupling between components of the system. This isolates the components and minimizes the system-wide impact of changes to the software or knowledge base and serves to retard architectural entropy. Data movement between components should be carefully managed within the architecture and the programmer should be constrained to remain within the system's architectural approach when performing maintenance. An additional architectural consequence of the need to provide modifiability is that the control flow for the system must be a visible and separate component of the architecture, which makes the architecture more complex. The task of knowledge engineering is complicated by modifiability because the knowledge acquisition effort must be more extensive, and can complicate the design because there must be a clean separation between the knowledge representation and the decision mechanism.

The need for *high fidelity* representations within the CGF affects the system's architecture because it must support multiple levels of fidelity in the representations of terrain, airframe, and human behavior. Regarding terrain, the system must have rapid access to different levels of detail so that the CGF is not burdened with reasoning about high detail terrain features that are outside of its sensor range. The airframe model, which encompasses vehicle dynamics models, sensor models, and weapon models, should be structured so that the level of fidelity in the computations dynamically adapts to the CGF's situation. For example, the highest fidelity should be used when the system is engaged in rapid maneuvers and lower fidelity representations are employed when the other components require additional computational cycles. However, the need for high fidelity conflicts with the need for real-time performance. The affect of this conflict can be ameliorated somewhat by the use of multiple levels of fidelity. As in the modifiability requirement, the demand for high fidelity increases the complexity and cost of the knowledge engineering task for the CGF. The implications of the human behavior modeling fidelity requirements for the system lie primarily in the areas of knowledge base design and decision mechanism design. For the knowledge base, the design should encapsulate related items of knowledge within a single access unit and separate unrelated knowledge components from each other. Here also there should be multiple levels of detail in the knowledge base so that the system can access information at the level of detail warranted by the world state and the available time before a decision must be made. This permits the decision mechanisms to atomically access the information they require and also permits the designer to update the knowledge bases with minimal impact upon other information in the knowledge base.

	Modifiability	**High Fidelity**	**Adaptive Decision Making**	**Automated Incorporation Of Experiences**
System Complexity	Increases system complexity due to the need to incorporate additional knowledge and implant a more flexible decision structure in the decision mechanism	N/A	Increases system complexity due to the need to implant an adaptive decision mechanism and a decision control mechanism	Increases system complexity due to the need to implant and operate an evaluation engine to determine how to score and incorporate decision results. Increases the complexity of the decision making component.
Software Architecture	Increases architectural entropy, control flow must be visible and traceable within the architecture	Architecture must be able to manage multiple levels of fidelity models	No global affect on the architecture, some effect on the decision making component	Mandates assembly of an evaluation engine to determine how to score and incorporate decision results
Real-time Performance	N/A	Competing requirements	Supports real-time performance	Competing requirements
Knowledge Engineering	Increases amount of work to be performed for knowledge representation development and separation of knowledge representation and decision mechanism	Increases amount of work to be performed for the development of the knowledge base and decision mechanism	Increases amount of work to be performed to elicit and represent basic principles for adaptability	Increases amount of work to be performed to design learning mechanism and capture experiential knowledge
Scalability	Complementary requirements	N/A	Attaining an adaptive decision mechanism requires scalability	Mandatory in the know ledge representation portion of the decision making component

Table 1: The Effect of CGF Requirements on Select Metrics of CGF Implementation

The requirement for *adaptive decision making* increases the complexity of the CGF's decision making component because the decision mechanism must robustly deal with incomplete information and uncertainty. The decision mechanism must be structured so that the amount of information considered when making the decision can be adaptively varied and so that additional possibilities can be considered as time and circumstances permit. Furthermore, the decision mechanism requires an external governor to limit the time and information resources consumed when computing a decision. Adaptive decision making supports achievement of real-time operation because it allows the decision making mechanisms to produce a viable answer with a specified amount of time and with limited information. As with the two previous requirements, this requirement increases the cost and effort of the knowledge engineering task. The requirement for high fidelity in the decision mechanism does not conflict with the requirement for the decision mechanism to be adaptable. At the system level, providing adaptable decision mechanisms provides the system with the means to perform scaleable decision making so that the amount of information considered and the time consumed by the decision can be varied dynamically according to the CGF's circumstances by the decision mechanism.

The requirement for *automated incorporation of CGF experiences* into its knowledge base and decision mechanism increases the complexity of the system because the system must be able to modify these two components during execution. In order to modify its behavior, the system requires an evaluation engine to assess the results of a decision, which increases the complexity of the decision making component and the knowledge base.

In light of these requirements, we developed a comprehensive CGF system architecture. We describe this architecture in the next section.

A SOLUTION

The requirements for an aircraft CGF, 1) modifiability, 2) high fidelity representations, and 3) adaptable decision mechanisms, are addressed within the current Automated Wingman (AW) project. In our architecture and design, we focused on the hierarchical structuring and modularity of both the processes and the knowledge to support modifiability, representation fidelity, and adaptable decision processing.

The Automated Wingman project addresses these requirements for aircraft CGFs in the virtual battlespace (Edwards, 1996). During operation, the AW flies in support of a lead, manned simulator with behaviors believable enough to be indistinguishable from human controlled entities. The AW responds to instructions from the flight lead and keeps the lead informed of any developing situations. In the event that the lead is destroyed, or the AW is separated from the lead, the AW determines its course of action in support of its mission and in light of it current status.

To achieve modifiability, knowledge engineering in the AW is predicated on straightforward incorporation of standard doctrine for strategy and tactics used with the various aircraft types. Representational fidelity is addressed through appropriate modeling and system representations used within the AW. To achieve decision process adaptability, we use a technique to allow decision making within an uncertain environment and employ an AW model representation that allows parameterization. In the following subsections we discuss the AW project, its CGF-specific architectural components, and the incorporation of AW into the Common Object Database (CODB) software architecture.

Automated Wingman and Fuzzy Logic

The Automated Wingman seeks to improve the state of the art for CGFs by using fuzzy logic as its core reasoning mechanism. Fuzzy logic provides the AW with the ability to reason with incomplete data and approximations, thereby enabling the AW to make decisions within an uncertain environment. The heart of the AW's reasoning mechanism is a fuzzy knowledge-based system that uses a hierarchy of knowledge bases in support of decision making. This provides the AW with a reasoning capability while the knowledge bases provide the information required to select appropriate tactics, to determine the required maneuvers to implement those tactics, and to fly the maneuvers. Fuzzy logic has already been shown to be successful to a limited degree in making simple decisions concerning movement of CGFs (Parsons, 1994).

Automated Wingman CGF-specific Architectural Components

Our motivation for the development of a general architecture to support the AW was to achieve system modifiability and form a basis for the design of broad classes of CGFs. Simply crafting CGFs by emphasizing differences is counterproductive and often results in only a few highly specialized types of CGFs being developed. Lack of a general approach to constructing CGFs will likely result in little or no information reuse between entity classes or even between entity types in the same class of CGF.

Our goal is to provide a general architecture for CGFs that naturally accounts for "variety" in a given type of CGF and presents a general approach for organizing and building vastly different CGFs, such as tanks versus aircraft. Our architecture consists of highly modular components where component interdependencies are well-defined and minimized.

Expanding system requirements often cause the CGF knowledge base and reasoning system to be modified and adapted to new system requirements throughout the life of the project and in the subsequently fielded system. As a result, the knowledge and reasoning components should be structured so that the knowledge base and reasoning system are disjoint, as in traditional knowledge-based systems. To allow for increased software adaptability, in our architecture the analysis and action components of the reasoning system are separate components as well. Furthermore, since we are using fuzzy logic, we implement each of these components as a hierarchy of objects that serve to aggregate information and dynamically limit the search space.

As discussed in the section detailing CGF requirements, the components of the airframe (aerodynamics model, avionics systems, and weapons packages) must be rapidly modifiable. Therefore, within the AW, these components are realized as separate objects that have a clean, robust interface to the remainder of the system. The reasoning components that use the outputs from the airframe CGF components are separate.

Our CGF reasoning mechanism consists of three components (Figure 2): a Pilot Skills Component (PSC), an Active Decisions Component (ADC), and a Physical Dynamics Component (PDC). The PDC encapsulates all the physical attributes and properties of the CGF. For example, in the AW, this component includes the aerodynamics model, entity-specific properties, aircraft capabilities, weapons load, sensors, damage assessment, and physical status. In addition, the PDC contains the processes for computing physical state changes, such as updating object position in the virtual environment. The PSC consists of those portions of the CGF that vary between individual entities within a type and class. This component serves to model the skills and ability of the pilot of an entity. For an aircraft entity, the constituents of the PSC consist of the pilot's ability to maintain situation awareness and to execute tactical and flight skills. These PSC components play an integral part in the decision making ability within the ADC. The ADC encompasses the intelligent decision making processes and the knowledge necessary to properly drive them. This includes the overall mission, goals and objectives, plan generation, reaction time, and crisis management ability, etc. Clearly, the ADC must accomplish many of its activities in real-time.

We separate these components from the remainder of the CGF architecture and from each other to insure that modifications are isolated and will not propagate throughout the entire system. The PDC is only responsible for the basic entity maneuver information, and functions completely unaware of the status of the other system components. Likewise, the ADC is solely responsible for decision making and only knows about the physical component's status based upon the data communicated in the system software architecture. The PSC is more closely tied to the ADC than the PDC because the ADC is responsible for computing control outputs for the entity based upon the modeled pilot's skills. The PSC supplies a description of the pilot's ability to the decision making component so that the decision can be appropriately constrained by the pilot's abilities. The division of capabilities between these basic components lessens the system level impact of any changes in the PDC, PSC, or ADC.

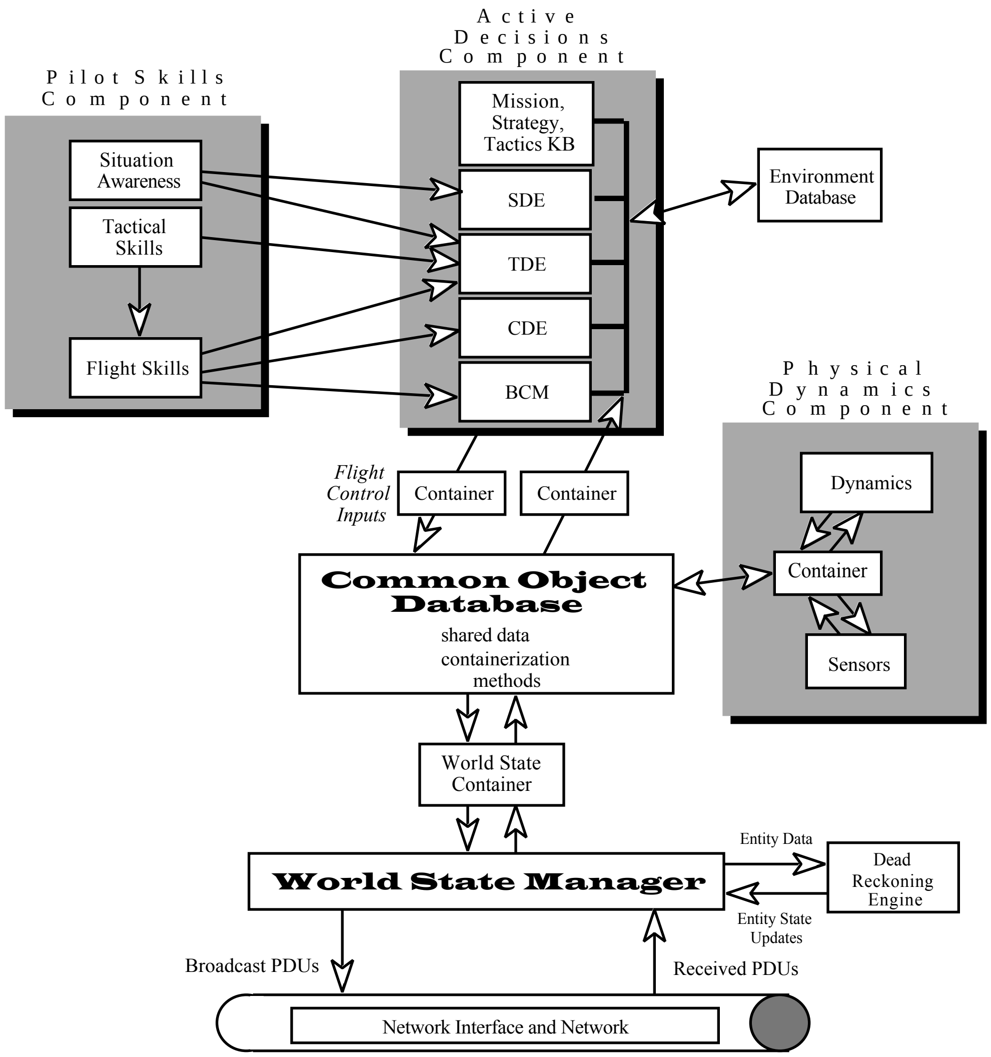

Figure 2. Automated Wingman System Architecture Incorporating the Common Object Database

The PSC and PDC contain all the information and status required to portray an aircraft and model its pilot's ability. The PDC encapsulates the entity state information and the PSC contains a representation for all the pilot skill variables. The key aspect of these two components is that these subsystems are completely parameterizable, and hence rapidly reconfigured and reused. We isolate entity control skills into the PSC because this separates the ability to parameterize the operator's capabilities from the decision making mechanisms used by the operator. Through this parameterization, any number of CGFs of a given type may be generated using a given ADC so that each entity has its own unique set of operator skills. The PSC models the pilot's skills as a hierarchy of capabilities. This scheme allows us to compose more complex skills from elementary skills and to compose the higher level skills using a careful weighting of the appropriate elementary skills. The drawback to this approach is that atomic skills must be carefully chosen and crafted so that high level skills have the desired performance. The PDC and PSC do not, however, perform decision making based upon the information they store. The decision making task is solely the responsibility of the ADC.

The ADC is the heart of the Automated Wingman, and holds the fuzzy logic decision engines. Decision making in the ADC is not based on a traditional goal-driven planning approach. Instead, the ADC contains a fuzzy planner that allows certain subgoals to remain unsatisfied but still have the supergoal satisfied. This decision making flexibility permits a much wider variety of possible behaviors and provides additional decision-making elasticity, allowing the CGF to achieve its mission in the face of uncertainty. That is, the system can tolerate uncertain satisfaction of subgoals and then use it as a measure. Also, the fuzzy approach provides a method for optimization when various subgoals are applicable but only one is desired. This use of fuzzy logic adds another behavioral distinction that can be exploited to create a diverse mix of entities.

Within the ADC, there are four primary reasoning modules of interest: the strategic decision engine (SDE), the tactical decision engine (TDE), the critical decision engine (CDE), and the basic control module (BCM). The ADC also contains relevant knowledge-bases specific to these reasoning modules. The SDE handles strategic matters related to accomplishing mission goals by continuously re-evaluating the completion status of mission objectives and re-planning to achieve the objectives in a deliberative fashion. To execute its plans, the SDE then requests the TDE to carry out the near term (tactical) objectives. The TDE operates under the direction of the SDE to manage near-term situations and determine a fine-grain course of action for imminent tactical situations. It then implements those actions as requests to the BCM. For example, for an aircraft, the TDE transmits stick and throttle settings to the BCM. The TDE is less deliberative than the SDE and must perform its functions in real-time. The CDE is a purely reactive reasoning system that deals with critical situations the AW might encounter. Its purpose is to enable the AW to survive a life-threatening situation, and it operates independently of mission goals and objectives. To operate effectively, the CDE monitors the world state until a critical situation is detected. The CDE then assumes control of the AW until the crisis has passed. During the crisis, the SDE and TDE monitor the AW's state so that they may resume control after the crisis has passed. Lastly, the BCM processes the requests of the TDE and CDE to pass as flight control inputs for the AW. Processing the requests takes into account the state of the PDC and PSC most relevant to the requests. For example, the BCM filters its flight control decision outputs using parameterized pilot ability ratings to execute a maneuver before it is applied to the aircraft's control inputs. Note that, due to the separable design for the ADC, the ADC could initially operate with only the BCM.

The above decomposition of the ADC maintains component independence. Furthermore, the knowledge-base decomposition mirrors that of the decision engines, allowing the various knowledge-bases to be constructed and tested independently. By modularizing our decision engines and the knowledge-bases in this fashion, traceability and certification of CGF behaviors are more easily achieved than in previous approaches to knowledge engineering CGFs.

The Common-Object Database

Our system architecture, Figure 2, relied upon the CGF requirements to guide the architectural definition. Figure 2 also illustrates the relationship of the complete software system architecture to the basic CGF-specific architectural components. Within the software architecture we use containers, which are data structures used to move large amounts of structured data between system components, to manage and control inter-component communication. The main AW components are specified as objects. These objects are the CGF-specific architectural components explained previously (PSC, ADC, and PDC), the Common Object Database (CODB), the World State Manager (WSM), and the Environment Database. Each of these objects are, in turn, hierarchically defined as a set of objects that use the containers to communicate with the other components of the AW via the CODB. The CODB holds all public data for the AW and all system components may access the CODB for data from other system components. The World State Manager is responsible for maintaining a complete description of the state of distributed virtual environment as communicated using DIS-formatted protocol data units.

To maintain isolation of the major components of the AW, we use containers to communicate all necessary information between the AW components and to all other external system components. The containers' size and format are fixed for the duration of an execution of the AW. However, because the contents of a container may be required by more than one other component, the containers transmit data from each component to the CODB rather than between components. The CODB holds all the exported data from all system components and functions as a routing mechanism for data movement between components. Each component updates its portion of the CODB via a container asynchronously. The only restriction we place on an update is that it can not occur while another component or the CODB is accessing the contents of the preceding container from the same component. The CODB also contains methods to repackage the information from several different component containers into a single component-specific outgoing container to a component. The CODB only holds data from the current containers. However, previous containers should be retained for later analysis. For this purpose, the CODB also maintains an Action History database. Due to the bandwidth required between the CODB and its components, we use the CODB architecture only for communication between system components that reside within a single host computer system. See Stytz for a more thorough discussion of the CODB (Stytz, 1996b).

SUMMARY AND FUTURE WORK

In this paper, we addressed the general problem of baseline requirements for aircraft CGFs. In light of these requirements, we presented a system architecture that satisfies these general requirements. We illustrated the application of this architecture using the AW project. By basing the AW architecture on the CODB approach, we achieved a flexible foundation for further development of aircraft CGFs while addressing the requirements for aircraft CGF development.

However, the set of requirements to be addressed by the architecture continues to change and evolve. Additional challenges come from the need to accommodate new types of data, such as environmental, radar, and infrared emissions. These data types are fundamentally different from the types of data currently processed in that they are variable throughout the region in which the phenomena are defined. We expect that the container approach to data flow within the software architecture will allow us to meet these new requirements.

The most challenging area of future work is the requirement for automated incorporation of experiences into the AW decision mechanisms and knowledge structure. We have not yet addressed this requirement because of the difficulty and system complexity incurred in the attempt. However, the AW knowledge structure, architecture, and decision mechanisms were designed and implemented with this requirement in mind. As the AW matures in its satisfaction of other requirements, we will address this requirement.

REFERENCES

Calder, R.B., Smith, J.E., Courtemanche, A.J., Mar, J.M.F., and Ceranowicz, A.Z. (1993) "ModSAF Behavior Simulation and Control," *Proceedings of the Third Conference on Computer-Generated Forces and Behavioral Representation*, Orlando, FL, pp. 347-356.

Edwards, M. and Stytz, M. R. (1996) "The Fuzzy Wingman: An Intelligent Companion for DIS-Compatible Flight Simulators", *The SPIE/SCS Joint 1996 SMC Simulation Multiconference: 1996 Military, Government, & Aerospace Simulation Conference*, vol. 28, no. 3, New Orleans, Louisiana, 77 - 82.

Giarratano, J. and Riley, G. (1994) *Expert Systems: Principles and Programming*, PWS Kent, Boston.

Kosko, B. (1993) *Fuzzy Thinking: The New Science of Fuzzy Logic*, Hyperion, New York, NY.

Laird, J. E., Newell, A., and Rosenbloom, P.S. (1987) "Soar: An architecture for general intelligence," *Artificial Intelligence*, vol. 33, pp. 1-64.

Laird, J. E., et. al.. (1995) "Simulated Intelligent Forces for Air: The Soar/IFOR Project 1995," *Proceedings of the Fifth Conference on Computer Generated Forces and Behavioral Representation*, pp. 27-36, Orlando, FL.

Parsons, J.D. (1994) "Using fuzzy logic control technology to simulation human decision-making in warfare models," *Proceedings of the Fourth Conference on Compute Generated forces and behavioral Representation,* pp. 519-529, Orlando, FL.

Schwartz, D. G. (1991) "A System for Reasoning with Imprecise Linguistic Information," *International Journal of Approximate Reasoning*, Vol 8, 463-468.

Stytz, M.R. (1996a) "Distributed Virtual Environments," *IEEE Computer Graphics and Applications,* vol. 16, no. 3, pp. 19-31.

Stytz, M. R., Adams, T., Garcia, B., Sheasby, S. M., and Zurita, B. (1996b) "Developments in Rapid Prototyping and Software Architecture for Distributed Virtual Environments," *IEEE Software*, vol. 12, to appear.

Tambe, M., Johnson, W. L., Jones, R.M., Koss, F., Laird, J. E., Rosenbloom, P. S., and Schwamb, K. (Spring 1995) "Intelligent Agents for Interactive Simulation Environments," *AI Magazine*, vol. 16, no. 1, pp. 15-40.

Zadeh, L. A. (1975) "The Concept of a Linguistic Variable and its Application to Approximate Reasoning," *Information Sciences*, Vol 8, 199-249 and 301-357.