Human and Organizational Behavior Representation in the Joint Synthetic Battlespace Experiment

Jeffrey Wallace
Barbara Hannibal
EnvoyTek, Inc.
San Diego, CA
jwallace@envoytek.com

Col. Phil Faye, USAF
Lt. Col. Emily Andrew, USAF
Electronic Systems Command CXC
Hanscom AFB, MA
phil.faye@hanscom.af.mil

David Skipper, Ph.D.
Kenneth De Tullio
Joseph McEniry
Bevilacqua Research Corporation
Huntsville, AL
davids@brc2.com

Keywords:
Human Behavior Representation, software development kit (SDK), HLA

ABSTRACT: *To date, computer-based virtual worlds that attempt to replicate human behavior have been plagued by a number of technical challenges. Most human behavior systems have been singular in nature and are customized to work with a specific simulation system. The Joint Synthetic Battlespace – Air Force (JSB-AF) effort during FY2003 – the JSB Experiment 1 – featured numerous improvements in military modeling and simulation across a variety of disciplines. One of the major achievements of the JSB Experiment was to demonstrate the ability to selectively enhance the modeling and simulation of a given phenomenology in a federation by integrating higher fidelity, or level of resolution, components. This was done for sensors, the environment, and human and organizational behavior modeling. Three main concepts were explored in terms of human and organization behavior representation. First, the JSB Experiment explored a software development kit (SDK) approach that permits intelligent systems and behavior representation to be embedded in different applications and simulation systems. Second, an integrative architecture for human and organization behavior modeling techniques was employed. In this way, the ability to simultaneously utilize a variety of specific human and organizational behavior representation techniques became possible, and allows the techniques to be selected based on the specific scenario or use-case requirements. Finally, the concept of a knowledge structure and execution engine that can be dynamically altered to model adaptive behavior was used, and we discuss the future implications of this capability.*

1 Introduction

Users are demanding high fidelity objects, interactions and environments. They want more speed with the higher fidelity, resolution and human behavior representation. Realistic synthetic representation, or stimulation of integrated systems, is desired. At the same time, they want to run multiple simulations with and without real-world systems using common models with shared results.

Various forms of simulation have been a mainstay technology for scientific and engineering disciples for many decades. From physics, to civil engineering, to operations research, the need to study, design, or predict the behavior of a complex system can often be satisfied through the application of simulation techniques. Over the past half-century, large bodies of knowledge and many software tools have been developed to aid in the design and analysis of simulations. Methodologies and software packages for domain specific needs, as well as more general solutions, have been developed. Typically these approaches rely on the application of a single software technology, such as imperative programming, and result in a software system whose sole function is as a digital simulation. This approach has been effective for simulation of isolated systems and loosely coupled federations of systems.

Over the last two decades numerous attempts have been made to develop large-scale simulations that span many levels of system resolution. The military models developed by operations researchers in support of defense department decision-makers are examples of comprehensive world representations of this variety. These efforts to develop simulations of large parts of reality, from switching circuits to human decision-making, have made use of most of what is known about software technologies and techniques. Large-scale, realistic model and simulation development also requires the application of a number of technologies. For example, declarative programming is often used to model human decision-making, while static portions of models are written in imperative languages for efficiency. However, most expert systems (declarative knowledge-based systems) are designed as stand alone packages and only provide loosely coupled interfaces to imperative languages. This is true of other software technologies and has hampered their general use to address modeling and simulation in a consistent and integrated manner.

Recent initiatives, such as the JSB-AF and the Joint Virtual Battlespace (JVB), have goals far beyond the use of large simulations as single-user tools. They have embarked on ambitious efforts at providing distributed simulations using virtual reality for training, virtual prototyping of hardware, and for analytical uses [5,6].

These ambitious, long-terms goals are unlikely to be met without a significant improvement in the underlying development environment and infrastructure for simulation software and it's relationship to human behavior representation systems and software. Similar difficulties exist in terms of the utilization of knowledge-based expert systems, and knowledge management software systems. Several daunting technical challenges still face military modeling and simulation developers. These challenges are:

- *Interaction Complexity* - the ability to permit clear, concise implementations among a diverse set of models that represent some significant portion of the real world.
- *Scalability* - the ability to efficiently execute over a wide range of simulation sizes, e.g. from 10 - 10^6 active simulation objects
- *Cognitive Process Modeling* - provide support for the modeling of human decision-making.

The JSB Experiment has chosen to begin addressing these challenges utilizing the following design and implementation guidelines:

- Employ an integrative architecture for human and organization behavior modeling techniques.
- Use a software development kit (SDK) approach for human and organizational behavior.
- Explore the usage of a knowledge structure and knowledge execution engine that can be dynamically altered to model adaptive behavior.

By utilizing an integrative architecture for behavior representation, the JSB-AF has the ability to simultaneously utilize a variety of specific human and organizational behavior representation techniques. This permits the user to select the human and organizational behavior representation techniques based on the specific scenario or use-case requirements. Thus, as behavior representation systems evolve that better address the interaction complexity, cognitive process modeling, and scalability challenges, the JSB-AF will be able to directly incorporate them and take advantage of the human and organizational behavior modeling improvements.

Perhaps more important is the realization that human behavior is a combination of distinctly different cognitive processes, types of knowledge, information, and data. Thus, situations involving different mental phenomena such recognition and strategy or problem solving behavior are in fact best modeled by different behavior modeling techniques. The ability to integrate different human and organizational models and simulations into composite, realistic representations is then a fundamental characteristic of any state-of-the-art approach to human and organization behavior modeling.

Utilizing an SDK for human and organizational behavior modeling in the JSB Experiment stems from the desire to investigate the ability to modify and upgrade the human and organizational behavior modeling in existing military models and simulations. In the JSB Experiment a variety of the chosen systems had existing behavior modeling that was to be replaced by different behaviors, or utilized as components of more complex behaviors. In general this approach permits intelligent systems and behavior representation to be embedded in different applications and simulation systems.

In the JSB Experiment, the SDK for behavior representation allows the development of component internal to the Joint Semi-Automated Forces (JSAF) application, and as individual application components (as HLA federates). This is consistent with the overall JSB Experiment approach to components and architecture. Composition is accomplished in two ways, as HLA federates and software development kits (SDKs) for different application domains such as environmental modeling [5,6] and human and organizational behavior representation.

This approach is consistent with Carnegie Mellon’s Software Engineering Institute (SEI) Attribute-Based Architectural Styles (ABASs) that promotes use of building blocks for designing software architecture by explicitly associating a reasoning framework with an architectural style [1]. This permits software developers are free to focus on constructing more robust applications for their customers. Using this strategy, the Joint Synthetic Battlespace Experiment Federation (JOSEF) demonstrates the use of a component-based development environment with inherent agility to flex to unanticipated customer needs for human behavior representation by employing an SDK approach. The SDK for behavior representation is based on an emerging ISO standard for knowledge representation, and is called the Cognitive Reasoning Engine (CORE) [7].

An overview of the JOSEF components will be provided next, along with an outline of which humans and organizations were modeled explicitly in the experiment. Next, a description is provided of the CORE toolkit. Finally we illustrate how the CORE toolkit was used to develop behavior models packaged both as standalone HLA federates and embedded within a complex simulation system as an integrated component.

2 Joint Synthetic Battlespace Experiment Federation Overview

There are many problems in current state-of-the-practice military models and simulations. In order to support efforts such as the Global Strike Task Force (GSTF), many improvements need to be designed and implemented. For example, there are very few dynamic and credible representations of sensors. Many current sensor models embed their own synthetic environments and human behavior models. In addition, these embedded environments are typically not dynamic. JOSEF separates the sensors from both the environment and human behavior representations. All sensors will use the same consistent and correlated environment. Additionally, the human and organization representations will be encapsulated in their own components, externally as a simulation federate or internally as a component based on a standard.

A key focus area of the JSB Experiment was to address the problem that most current synthetic environments in general are not dynamic and are not correlated through the entire electromagnetic spectrum. In order to solve this deficiency, JOSEF's Common Synthetic Environment (CSE) provides a common, correlated and integrated environmental representation covering the EO/IR/RF part of the spectrum. In addition, many synthetic environments do not have the capability of representing dynamic signatures for the entities contained within them. The CSE provides dynamic signatures for the entities contained within a scenario. These dynamic signatures are influenced by ephemeral conditions (Sun position, time of day, etc…) and in future experiments by vehicle states. Other hard to model phenomenology, such as RF ducting, and the ability to produce high-resolution geo-specific terrain and feature modeling are also addressed.

In order to make scenarios realistic, various types of clutter must be represented. Many current synthetic environments inconsistently represent object clutter, and JOSEF will provide a varied clutter environment for accurate and dynamic stressing of the sensor and decision models. A related issue is the ability of a synthetic environment to provide decoys and false targets that represent a realistic battlefield. The CSE will provide decoys and false targets, which will stress the sensor and decision models so as to give a more realistic representation of Command, Control, Intelligence, Surveillance, and Reconnaissance (C2ISR) processes and in particular specific sensor capabilities.

Most military models and simulations, with the exception of limited performance analysis studies, do not account for process, from the perspective of command and control (C2) decision functions, and processing (from the perspective of C2 systems) latencies and workloads representations. Many current simulations don't capture battlefield process latencies and the human workload representations. JOSEF captures and models these latencies by modeling delays at the appropriate point in C2 systems and decision processes, and workloads through human and organizational behavior representation. This capability is used to model operators, decision makers – in particular the Time Sensitive Targeting (TST) Cell.

Other C2 related phenomenology, such as data "gridlock" and errors in message transmissions in combat communications systems need to be represented. Again, most modern simulations don't model this gridlock phenomenon and its causes. Although JOSEF will not model all aspects of this phenomenology in the initial experiment, it will incorporate the necessary interfaces and capabilities to model this phenomenology.

The JOSEF components are shown below in Figure 1:

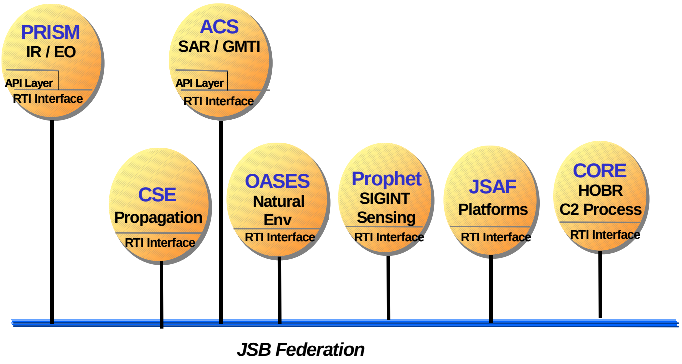

Figure 1. JOSEF components.

The JSB Experiment employed JOSEF to model a Time Sensitive Targeting (TST) vignette where the TST Cell directs a variety of ISR assets to produce mensurated target coordinates to pass on to strike assets. The TST Cell model was produced at exercise the rapid prototype capability of the CORE toolkit to model the specific TST Cell process of interest. In addition, there are several human operators controlling platforms and sensors present in the scenario vignette to be modeled in JOSEF. They are:

- UAV Operators
- F-15E Operators
- JSTARS Operators

The implementation details of will be discussed later, but the TST Cell and JSTARS Operators were standalone HLA federates and the UAV Operator and F-15E Operators are implemented as internal components in JSAF. A discussion of the CORE toolkit is provided, and the key elements of the implementation of the JOSEF behavior components provided.

3 Cognitive Reasoning Engine (CORE) Overview

Recent advances in detailed conceptual modeling have been shown to produce more realistic behavior in computer-generated agents [1]. While these advances can provide improved realism in the training experience, they can also provide the means to significantly improve the efficiency of operational military systems and military system design and engineering.

Current agent-based approaches to knowledge handling rely heavily on classic representation techniques that were developed to be used with older linear programming languages such as BASIC and FORTRAN. Each of these techniques, which include rule-based expert systems, genetic algorithms, neural networks, and similar approaches all suffer from one or more limitations that make them individually unsuitable to meet the needs of future intelligent software applications.

Recently, however, a new approach to knowledge representation, handling and distribution has been introduced that offers a solution to the older methods. The Cognitive Reasoning Engine (CORE) toolkit uses conceptual graphs as the basis for its knowledge representation scheme. Conceptual Graphs (CGs) are based on the work of John Sowa [8], who in turn based his work on a combination of semantic networks and the graphical logic of Charles Peirce [9]. With a foundation in logic, CGs represent a first-order logic model of a system. Combined with the defined logic operations from Sowa and Peirce, the CORE graph processor can now performs inference procedures on the model itself. While this property is certainly essential, there are other aspects of CGs, such as application integration and program adaptation that represent a significant step forward in knowledge utilization and management.

Visually, a Conceptual Graph mimics the knowledge representation ability of common diagrams used in discussions using whiteboards, slides, or even table napkins. These drawings are often text snippets (typically enclosed in squares or ovals) and lines (possibly with a label) connecting one snippet to another. Experts often use these visual aids to quickly and effectively communicate complicated, technical details during brainstorming sessions. In CGs, text snippets, in a square, are called Concepts. The line connections are enhanced, with ovals containing more text, to represent complex relations between various concepts in a manner entirely consist with common "brainstorming picture". A sample CG is shown in Figure 2 below.

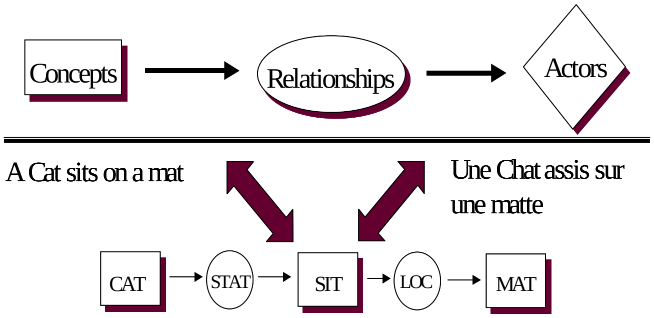

Figure 2. Conceptual graph elements.

Structurally, CGs provide advantages for modeling complex behaviors. The entire structure is inherently hierarchical, so that the analyst can decompose concepts to the appropriate level of detail needed to describe the phenomena at hand. The hierarchical method of expression also permits operation at higher, more aggregated levels when it is beneficial to do so. This nesting also provides contextual basis for an entire system or a specific concept. The structure is a general representation that can capture any aspect of the system.

For instance, uncertain knowledge can be explicitly represented by including the concepts of belief and associating the beliefs with given components and values in a manner similar to belief networks or fuzzy valued expressions. Similarly, other forms of imperfect knowledge can be directly represented and manipulated to as a component of a logic model.

Utilization of knowledge models represented by CGs also provide advantages over stand-alone approaches. Ordinarily, combining procedures with declarative knowledge into a hybrid system is performed by force fitting the two technologies. However, CGs provide functionality as a fundamental part of the representation.

The role of actor types is to provide a way to encapsulate interfaces to other functional APIs. CGs simply represent a functional relationship as a diamond shaped symbol that indicates that some functional activity takes place within.

This powerful approach allows easy integration of functions (imperative programming) and logic (declarative programming) in the same representation, so that those parts that are well understood and easily represented as mathematical, or other computational approaches, can be bound into the logic system and the correct point. These actors also add the mechanism to connect a logic system to the "real world". That is the actors can form interfaces to the outside as well as providing active relationships within the CG.

Practical implementations of CGs require two important developments. The first is a means of extending the actor set by the end user to meet special situations without the need to go back to the original developers for a new version of the graph system. The second is an embeddable graph processor that is able to read a representation, implement it inside of a users application, along with the required interface actors, then to process the CG in an appropriate manner in conjunction with the rest of the users application. The first development has been demonstrated in the JSB system and the capability is being refined for the next release of CORE. The second development has been shown in numerous applications and it is now also being extended for the next release of the CORE toolset.

Knowledge interchange is the last, but very critical component needed to build large complex behaviors in an affordable manner. Although the arguments for knowledge interchange have been repeated since the early 1990s, the practical issues surrounding that goal have delayed its realization. Previous efforts have included the Knowledge Interchange Format (KIF) from Stanford University and the Conceptual Graph Interchange Format (CGIF).

Both of these efforts have been under scrutiny for ANSI standardization. However, the international community has begun to realize the significance of knowledge representations and interchange. Consequently, at the recent Open Forum on Metadata Registries (Santa Fe, January 2003), ISO planned to open a work item on Common Logic, in which KIF and CGIF would represent specific interchangeable implementations of a potential international standard. While this has effectively brought the ANSI standardization efforts to a halt, the potential for an international standard is a significant step up in utility for any representation.

The ability to represent knowledge in a graphical form allows agents and systems to be readily grasped by the human observer, which in turn leads to much more maintainable systems of knowledge than older rule-based methods. As a parallel, object-oriented schema, the representation is explicit about interactions, which leads to a non-brittle system. There is a further significance for the military. For example, knowledge can be added or taken away from a conceptual graph-based knowledge base without interfering with the logical connections of the other objects present. This opens the door for providing adaptable knowledge bases and in the future, learning.

The Government currently has a free-use license for the CORE Toolkit that allows the creation and maintenance of conceptual graphs. The toolkit includes a library to include with an application that performs the parsing and execution of conceptual graphs to control the application.

3.1 HBR and CORE

Human Behavior Representation (HBR) in M&S requires speed, efficiency, and the capability to store and retrieve deep levels of knowledge, while also providing scalability, adaptability, and low maintenance costs. Further, many HBRs are ad hoc combinations of logic models and functional or empirical models that describe various aspects of behavior. Based on these needs and the descriptions in the previous section, CGs appear to offer the opportunity to encompass and improve the existing behavior representations by addressing multiple shortcomings of traditional approaches, such as:

- Brittleness
- Limited scalability
- Limited adaptability
- Mixed Representations
- High Maintenance
- Slow performance

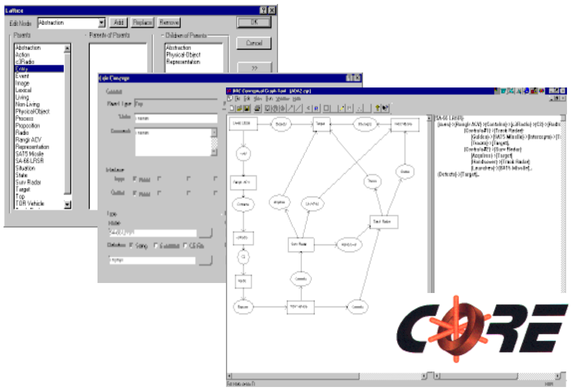

Figure 3. CORE Graphical Modeling Interface.

As the user inserts the knowledge into the system, shown in Figure 3 above, it is automatically parsed into ANSI Conceptual Graph Interchange Format (CGIF), the new national standard for knowledge interchange. This allows knowledge to be easily shared and distributed among users and other applications conforming to the standard.

4 Using CORE in the JSB Experiment

As was stated previously, there are several human operators and an organization present in the scenario vignette to be modeled in JOSEF. They are:

- Time Sensitive Targeting Cell staff
- JSTARS Operators
- UAV Operators
- F-15E Operators

The TST Cell modeled the activity of asset tasking and coordinate mensuration as an aggregate organization. An optimization heuristic was implemented that sought to minimize the time to mensurated the coordinates of a target from the time the target was first detected. This behavior was encapsulated as an HLA federate that received target locations via HLA interactions and issued asset tasking and execution messages via HLA interactions.

The JSTARS Operators were also implemented as an HLA Federate, and received asset tasking messages, controlled the GMTI/SAR sensor based on those messages, and then generated target report messages. This behavior was rapidly prototyped due to the need to interact with an engineering level GMTI/SAR model and integrate platform information from a separate HLA federate.

The UAV Operator and F-15E Operator behavior were built as internal components to JSAF, where the CORE graph processor engine was linked into the JSAF executable and the actors in the CGs made calls to JSAF APIs to:

- Control the platforms which were modeled in JSAF
- Send and receive HLA messages related to asset tasking and target reporting to and from the TST Cell
- Control the engineering level models of the EO, IR, and SAR sensors on the platforms – which are all in separate HLA federates.

In the case of the UAV Operator, a call is also made to a behavior model that represents a human looking at an EO/IR sensor display and picks out a target from a cluttered background. The ability to integrate behaviors from both JSAF and a target recognition/vision model demonstrates the ability of CORE to integrate a collection of behavior models into a realistic composite behavior.

The Time Sensitive Targeting Cell, JSTARS Operators, UAV Operator, and the F-15E Operators are modeled using the CORE Conceptual Graph editor and the CORE graph processor engine. In terms of rapid behavior modeling, the CG notation of concepts (rectangles), relationships (ovals), and actors (diamonds) provides us with a visual programming capability that can not only express rules and facts, like traditional expert system shells such as CLIPS, but also compiled libraries of behavioral code embedded in actors. This practical and efficient mechanism for building hybrid knowledge and behavior agents is illustrated below for the JSTARS Operators.

4.1 Joint Stars (JSTARS) Operators

In order to illustrate more fully the CG technique, a collection of several of the CGs used in developing the JSTARS Operator representation are discussed. The implementation of the agent to model the JSTARS Operators is slightly less complex than for the UAV and F-15E Operator models in that no maneuvering needs to be performed, since the JSTARS is flying a fixed figure eight racetrack. This orbit is located so that the target areas of interest are within the APG-8 radar system, used in the SAR and GMTI modes. The bulk of the work in this implementation is determining when to activate the sensors based on where the JSTARS is on the racetrack, and doing the HLA coordination with the HLA federate that models the sensor in terms of commands and responses – in this case a list of detections.

In Figure 4, the TST Cell concept represents an encapsulation of the "ghost" representation of the TST Cell and hides the HLA calls made to get information from the TST Cell federate. As with the established pattern, the JSTARS receives a task order in order to initiate execution of the agent CGs.

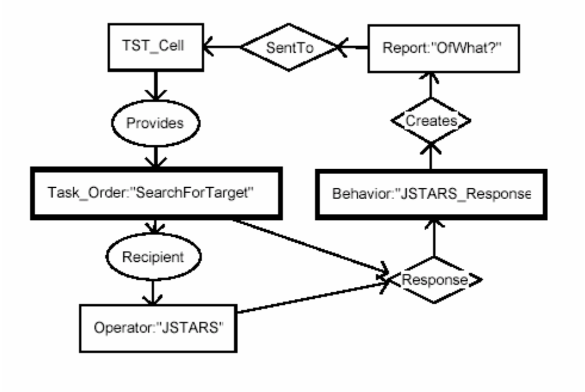

Figure 4. Top-level CG for the JSTARS Operators.

Synchronization is enforced before the Operator responds using an actor. When both inputs to the response actor are TRUE, namely a task order has arrived and the Operator is ready to receive it, the Response concept is then evaluated. When the Response CG has been executed, a report is constructed and sent to the TST Cell.

The Response CG for the JSTARS, illustrated in Figure 5 below, differs from the UAV and F-15E implementations, but has one fundamental structural similarity to the UAV and F-15E Operator Responses. The Response CG has two parts, the top CG governing an action that must be performed before the sensors can be used with the key action being the computation of the values of the flag “SAR Done”. This represents the integration of rule-based declarative systems with imperative systems.

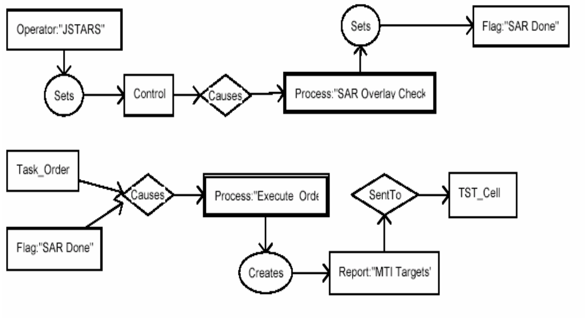

Figure 5. JSTARS Response CG.

The **SAR Overlay Check** concept in figure 5 is implemented below in Figure 6. This demonstrates the ability to hierarchically represent complex knowledge structures using CGs. The key function performed is the encapsulation of the HLA interaction with the ACS federates that models the APG-8 in order to generate a SAR patch map. This sets the “Exists” and “Time Current” flags to be true and allows the rest of the processes controlling the use of the APG-8 sensor in the GMTI mode. This synchronization must be employed to generate the real-world behavior that governs the cueing of the sensor between the GMTI and SAR modes.

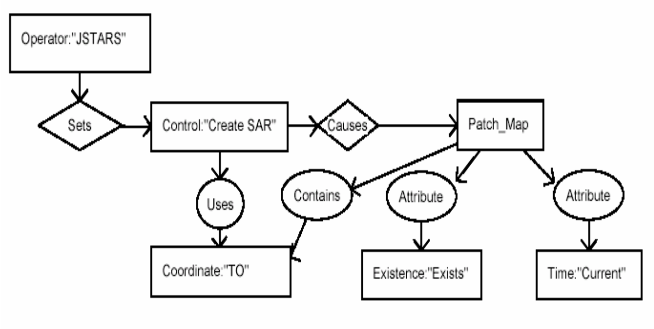

Figure 6. The SAR Overlay Check CG.

As is shown in Figure 7 below, once these synchronization points in the previous CGs are passed, the **Scan Area** concept, implemented as a CG in Figure 8 can be executed.

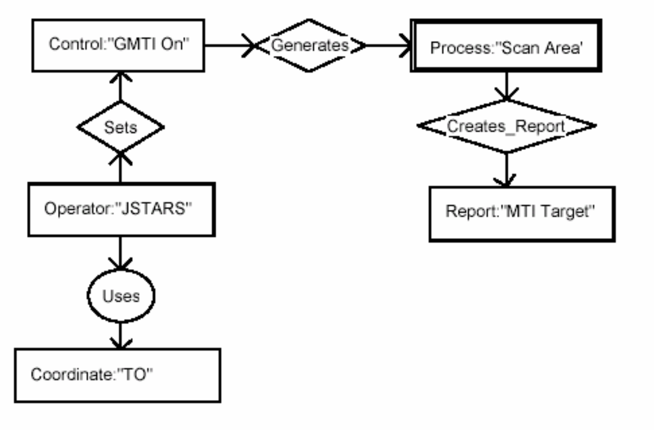

Figure 7. GMTI Control CG.

The most important features in the **SAR Scan Area** CG are the way the HLA calls made to the ACS RF Sensor federate and the data received from it are nicely hidden using the semantics above – “causing” the scan and “generating” the list of targets. Note that unlike the situation in the previous section on UAV Operator modeling, the code that evaluates the output of the sensor is still bundled into the RF sensor federate, unlike the way the EO federate separated sensor functions and human behavior, in this case operators looking at the GMTI display and making detection decisions.

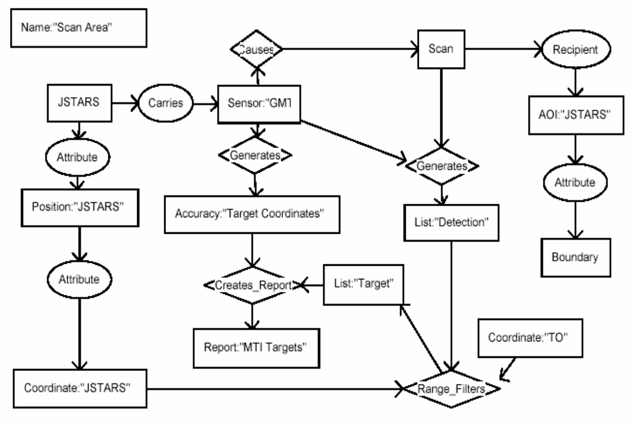

Figure 8. JSTARS Scan Area CG for GMTI.

4.2 UAV Operator Modeling

A brief overview is provided in this section outlining the modeling that is required to implement the UAV Operator for the purposes of the JSB Experiment. This behavior was embedded in JSAF, and calls JSAF behaviors to implement platform and sensor control.

4.2.1 Maneuvering

The on-station location and pattern consists of two minute legs and a standard turn at each end – a thirty degree bank turn, all done at standard loitering speeds (approximately 70 knots). The wind speed and direction will determine the power setting required to hold a constant speed. The focal point for pattern exists inside the Camden Ridge/Pine Hills MOA.

The behavior governing the transit to and from the on-station location to perform tasks, and re-tasking behaviors is as follows. When the UAV operator gets a coordinate, the operator will first point the camera at the coordinates. The UAV will then be commanded to fly at the coordinate. The UAV will fly at target, and if no detection is made, fly outbound for 2 minutes and then do a 60-degree bank turn and head back toward the coordinate. While the maneuver is being performed, the operator moves the camera in a search pattern.

When given coordinates to search, the following pattern is flown while searching. The sensor is cued to the target coordinates and the UAV is flown towards the coordinates. The coordinate refinement operation – flying right at something is performed until an over flight has occurred.

4.2.2 Controlling the Sensor

When a search area has been determined, the sensor is controlled in the following manner while transiting to the search area. The sensor is cued, and the operator will wait until he has some resolution in the target area. The operator will not search the target area until he is within 10 miles. Roads are searched first if present and near the target coordinates. If the target is not in the Sensor field-of-view (FOV), scan right to left to two times the sensor FOV to each side, as indicated in the figure below.

If the target is not found, hit recue button, which send the FOV center back to the original target coordinates. When the sensor is covering an area, a raster search is performed. A raster search is an outward expanding rectangular pattern, starting from the target coordinates expanding by 0.5 FOV each time.

5 Conclusions and Future Work

The goal of the JSB Experiment was to implement and architect JOSEF as a step in the direction of maturing Human Behavior technology to support simulation environments. This was accomplished by using the approach to component-based design via integrative architecture and the SDK approach used in other areas of the JSB Experiment, such as environmental modeling and the development of the Common Synthetic Environment.

The ultimate goal of the JSB-AF would be to support and address:

- Sustained operations, which will enable intelligent agents to affect personnel reductions.
- Environmental ambiguity with respect to methods for collaboration and knowledge reuse.
- Information overload as intelligent agents assimilate information into knowledge.

6 References

[1] Carnegie Mellon University, "Attribute based Architectural Style, 2001. http://www.sei.cmu.edu/ata/abas.html

[2] Defense Modeling and Simulation Office (1998a) “Department of Defense High Level Architecture Rules, Version 1.3,” February 5 1998, Online document at http://hla.dmso.mil.

[3] Defense Modeling and Simulation Office (1998b) “Department of Defense High Level Architecture Object Model Template, Version 1.3,” February 5 1998, Online document at http://hla.dmso.mil.

[4] Defense Modeling and Simulation Office (1998c) “Department of Defense High Level Architecture Interface Specification, Version 1.3,” April 2 1998, Online document at http://hla.dmso.mil.

[5] Faye, Col Phil, USAF, et al., 2002. “The Joint Synthetic Battlespace for Simulation-based Acquisition: A Steptowards the Future of Military Modeling and Simulation.” In *Proceedings of the 2002 Interservice and Industry Training, Simulation, and Education Conference*, NTSA, Orlando, FL, December 2-6, 2002.

[6] Electronic Systems Command/CXC, “JSB Experiment Design,” Hanscom AFB, MA, April 2002.

[7] Skipper D., et. al. 2000, “Implementing Knowledge Interchange for Simulated Entities.” In *Proceedings of the 2000 Huntsville Simulation Conference,* Society for Computer Simulation, San Diego, CA, Oct 2000.

[8] Sowa, J.F., Information Processing in Mind and Machine. 1984, Reading, MA: Addison-Wesley Publ.

[9] Peirce, C.S.: Collected Papers of Charles S. Peirce, Hartshorne, Weiss, and Burks (eds.), Harvard University Press, Cambridge, MA, 1932.