9th International
Command and Control Research and Technology Symposium
June 15-17, 2004
Loews Coronado Bay Resort
San Diego, CA

TOPIC: Information Age Transformation

TITLE: IEIST Guardian Agent and Force Template Technologies Provide Warfighter Connectivity to the Global Information Grid

AUTHORS:

(1) Charles P. Satterthwaite (**point of contact**)
Air Force Research Laboratory
Information Directorate, Information Technology Division
Embedded Information Systems Engineering Branch (AFRL/IFTA)
2241 Avionics Circle, Bldg. 620
Wright-Patterson AFB, OH 45433-7334
Phone: 937-255-6548 x3584
Fax: 937-656-4277
Email: charles.satterthwaite@wpafb.af.mil

(2) David E. Corman
The Boeing Company
P.O. Box 516 MC S111-1335
St. Louis, MO 63166-0516
Phone: 314-234-3725
Fax: 314-233-8323
Email: david.e.corman@boeing.com

(3) Eric J. Martens
The Boeing Company
P.O. Box 516 MC S111-1335
St. Louis, MO 63166-0516
Phone: 314-233-0086
Fax: 314-233-8323
Email: eric.j.martens@boeing.com

9th International
Command and Control Research and Technology Symposium
June 15-17, 2004
Loews Coronado Bay Resort
San Diego, CA

Charles P. Satterthwaite
Air Force Research Laboratory
Information Directorate, Information Technology Division
Embedded Information Systems Engineering Branch (AFRL/IFTA)
2241 Avionics Circle, Bldg. 620
Wright-Patterson AFB, OH 45433-7334
Phone: 937-255-6548 x3584
Fax: 937-656-4277
Email: charles.satterthwaite@wpafb.af.mil

Dr. David E. Corman
The Boeing Company
P.O. Box 516 MC S270-4265
St. Louis, MO 63166-0516
Phone: 314-234-3725
Fax: 314-233-8323
Email: david.e.corman@boeing.com

Eric J. Martens
The Boeing Company
P.O. Box 516 MC S270-4265
St. Louis, MO 63166-0516
Phone: 314-233-0086
Fax: 314-233-8323
Email: eric.j.martens@boeing.com

Abstract

Thirteen years after Desert Storm, conduct of Time Critical Target (TCT) operations remains one of the most difficult challenges facing US military forces. While a variety of DoD and DARPA programs are addressing technologies to locate and identify TCTs, finding the target candidate is only one part of the problem. Success will only occur when we shorten the entire "kill chain", and operate within the enemy's maneuver timeline. The automated exchange and processing of battlefield information is critical to achieving viable decision timelines in this arena. The situation demands a secure, robust network backbone supporting automated decision aids designed to execute commander's guidance. Critical decision aids include the ability to monitor and exchange critical tactical information, to evaluate real-time intelligence and generate actionable Target Evidence Files and to re-assign en-route tactical and support assets to higher value tasks.

The Air Force Research Laboratory (AFRL), with support of The Boeing Company, is executing several research initiatives targeted at these information exchange shortfalls. AFRL is developing the Joint Battlespace Infosphere (JBI) as a means to realize information dominance. In effect, the JBI can be viewed as a tactical Internet that provides unprecedented access to data sources. Through this wide-area network connectivity, the JBI can be accessed, searched, and manipulated to create new products.

This paper will discuss the Insertion of Embedded Infosphere Support Technology (IEIST) research in which the Guardian Agent (GA) will be embedded within the Force Template (FT) and transmitted to the appropriate C^2 node(s) over a tactical data link. The demonstration C^2 node is planned to be the Advanced AWACS Prototype Software, which will host the transmitted GAs as well as the TCT Evidence File Generation Agents, and the Real-time Battle Management system that will match tactical assets to selected TCTs.

Introduction

According to Air Force vision statements Joint Vision 2010 and Joint Vision 2020 [1, 2]; the future execution of airborne missions will differ significantly from today in terms of force composition, operational concept and most importantly reliance on information superiority. The heritage of the Air Force has been air superiority in terms of system capabilities. Many of today's weapon systems are approaching 20 or more years in age. Today most missions are planned using less than the latest intelligence information. As a result, weapon systems are often forced to respond to threats as they are encountered during the course of the mission, even though in many instances threat movements had been detected by intelligence systems but the information had not yet reached the operational mission planning systems.

Future airborne missions will look very different. Air superiority is still required, but it will be achieved with very intelligent weapon systems, equipped with the latest technology, and plugged into the latest information. In other words, these missions are globally aware, can globally engage, and can dynamically respond to their commander's guidance [11, 12].

In order for this new concept of warfighting to become a reality, technical innovation will be required in many areas. Some of these areas include: distribution and access of Global Information; rapid re-configurability of weapon systems and their supporting systems; and inter-operable capabilities between the employed weapon systems.

A huge step forward in realizing this technical innovation is through the Insertion of Embedded Infosphere Support Technology (IEIST) program. IEIST offers two unique technical innovations that give weapon systems access to global information, allows these systems to be coordinated and paired with time critical targets (TCTs), and allows these systems the capability of better leveraging each others resources. These innovations are the Force Template and the Guardian Agent. The Force Template (FT) characterizes the weapon system so that it has a standard way of communicating with elements of the

Global Information Grid (GIG). The FT knows the particular details of its weapon system such as its weapon load, available fuel, location, operational capabilities, crew experience, and interface requirements. The Guardian Agent (GA) is a unique way of adding capability to the weapon system, without straining available onboard resources. The GA lives off-board the aircraft (probably on the controlling C^2 aircraft). The GA is created from the weapon systems' FT, and is unique to its weapon system. It can communicate with other GAs when they have important contributions for its weapon system, but has no dependency on these other GAs or any particular information source. The GA subscribes to information for its weapon system when and only when the information is pertinent to that system's mission. It filters extraneous information, and formats pertinent information for timely aircrew display, at the aircrew's request. The GA also serves as a proxy for publishing information from the weapon system to the GIG or more precisely the JBI.

Relevance to C^2

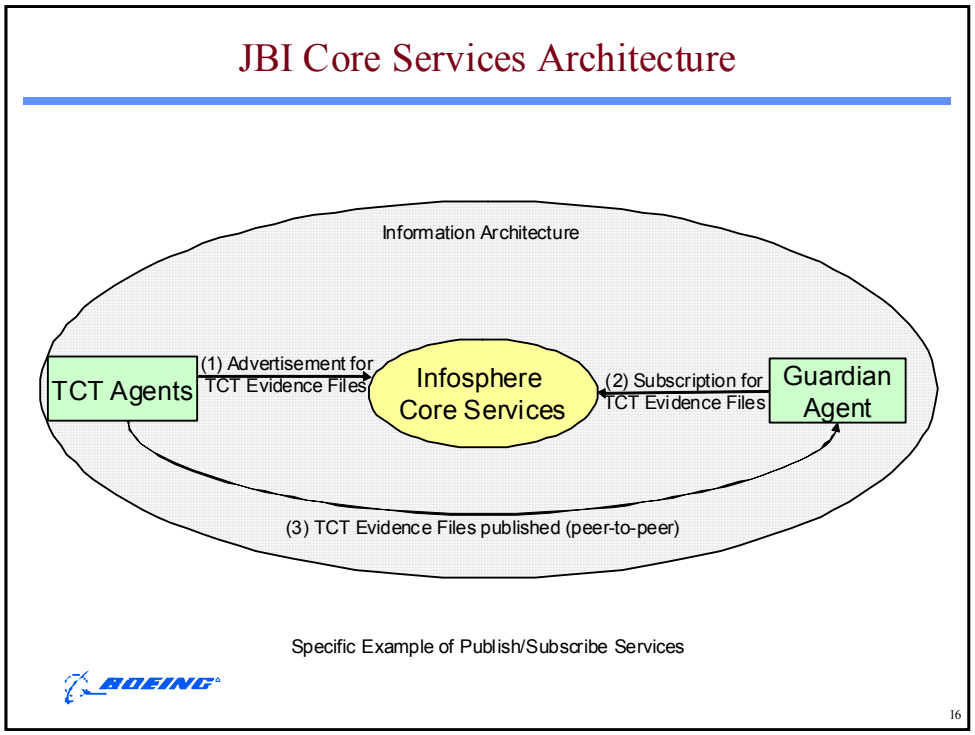

Figure 1 – JBI Core Service Architecture

Whilst the ability to access quantities of data is vital, the essential capability of the JBI is to support the translation of data into actionable information (Figures 1 & 2). This capability directly satisfies the principal need of Command and Control. The AFRL/Boeing IEIST initiative has already demonstrated dramatic improvements in the exchange of information between deployed tactical elements including airborne C^2 and information nodes worldwide (Figure 3).

IEIST focuses on the integration and requirements for off-board software agents, designed to augment embedded tactical systems and plug into the evolving JBI, while still providing interoperability with legacy systems and communication links. The essence of IEIST is understanding the information needs and collection capabilities of the platform and matching these against information sources and destinations in the JBI. Key elements of the IEIST Architecture include the Guardian Agent (GA), the Host Agent (HA) and the Force Template (FT). The GA identifies and accesses information of interest across the JBI, evaluates the tactical utility of the accessed information, and transmits the information to the tactical element (aircraft) using available communications. The HA is a thin layer, which resides on the tactical node and operates in conjunction with the Operational Flight Software. The Host Agent provides an interface between extant tactical systems and Guardian Agents, using legacy tactical data links for communications. The FT is an Information Object that defines the information generation capabilities and information need of the tactical platform. IEIST has already demonstrated integration of GAs and HAs for multiple tactical assets and C^2 nodes communicating using JBI protocols and services over a simulated Link 16 interface. Other agents within the IEIST demo scenario have automatically generated TCT Evidence Files, which were transmitted to and exhibited on the cockpit displays of assigned prosecution assets.

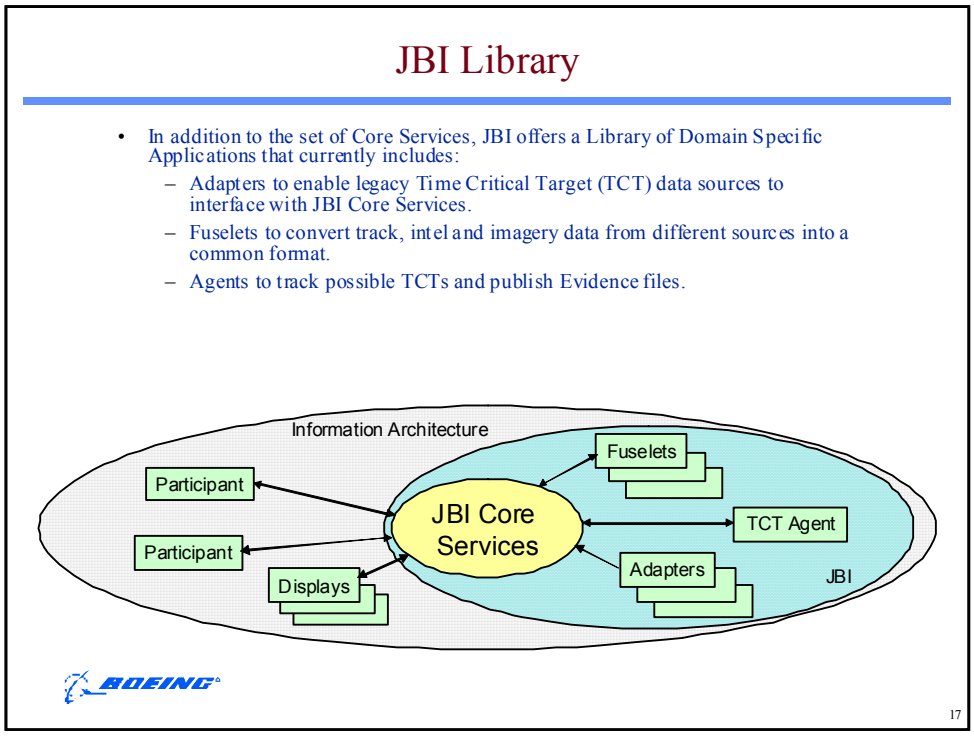

Figure 2 – JBI Library

Guardian Agent

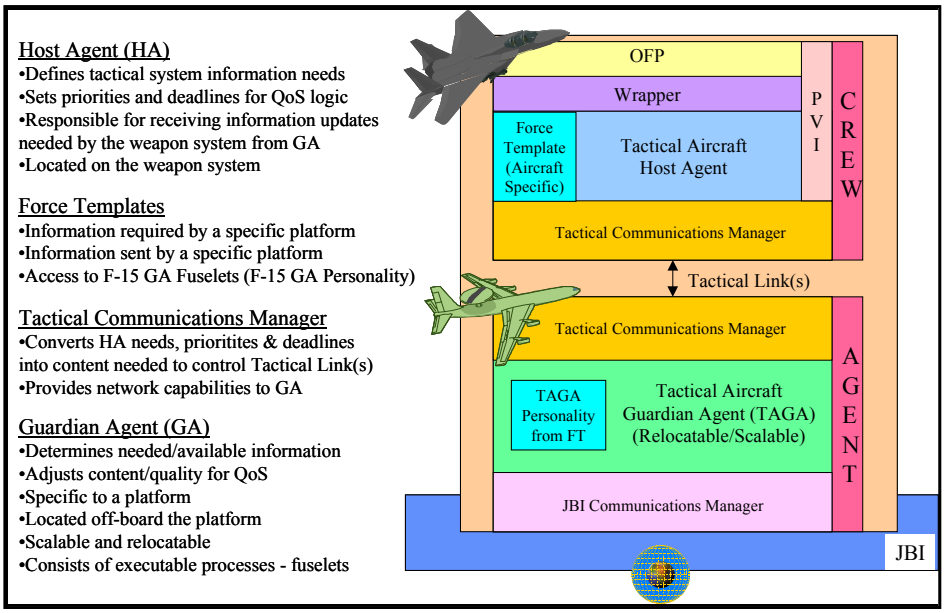

Figure 3 – Tactical Elements And Information Nodes

The Joint Battlespace Infosphere (JBI) will provide "never before available" information to tactical platforms. The JBI will also provide Command and Control operators and commanders real-time situational awareness of dynamic changes to the battlefield. From the tactical platform view, IEIST enables the platform to gain access to information that can magnify its survivability and effectiveness. From the C^2 view, IEIST can enable the legacy platform to become a smart resource for the C^2 commander.

The JBI can provide sensor data on "over the horizon" threats to the legacy platform. These are threats that lie along its mission route and in which they would not normally be detectable by the platform's sensors until late of the mission. With the JBI, the platform can gain access to sensor reports from other platforms providing coverage along its route. The JBI allows every sensor in the battlefield to become virtual sensor for the platform. The JBI can also provide timely information (such as intelligence images) that can affect its mission. JBI information will enable the tactical platform to become more survivable and lethal. From the Command and Control viewpoint, it is desirable in a dynamic battlefield to have access to "smart resources" that can have their missions dynamically changed or provide dynamic information. The JBI provides the infrastructure for implementing this capability.

There are several issues to be resolved in order to implement the legacy tactical platform into a JBI. First, there is the issue of avoiding information over-load for the pilot or platform operator (in the case of a UAV). The legacy platform has other constraints when connecting to a JBI. Typically, there is limited excess processing and memory available on the platform for implementing smart software that can filter and process information going to and from the JBI. There is also limited bandwidth available for use to between the tactical platform and the JBI C^2 node. Transmitting all information available within the JBI for evaluation on-board the platform could easily overwhelm the available bandwidth.

To address the above restrictions and constraints, the concept of a Guardian Agent has been developed by the IEIST program. The Guardian Agent is an intelligent agent that serves several roles in connecting a legacy platform to the JBI. To over-come the limitations of available processing power on the platform and the limited bandwidth, the Guardian Agent is located on a high capacity server in a C^2 node. The Guardian Agent is connected to the JBI via this C^2 node server. The Guardian Agent also connected to the legacy platform via a standard tactical data link such as Link-16. Figure 4 shows the IEIST architecture and how the Guardian Agent fits into this architecture.

A small software component called the Host Agent resides on the legacy platform. The Host Agent provides the limited required functionality for implementing the link between the Guardian Agent and the legacy platform's computer and/or pilot interfaces. The Host Agent can be thought of as the localized "thin client" for the much more capable intelligent Guardian Agent.

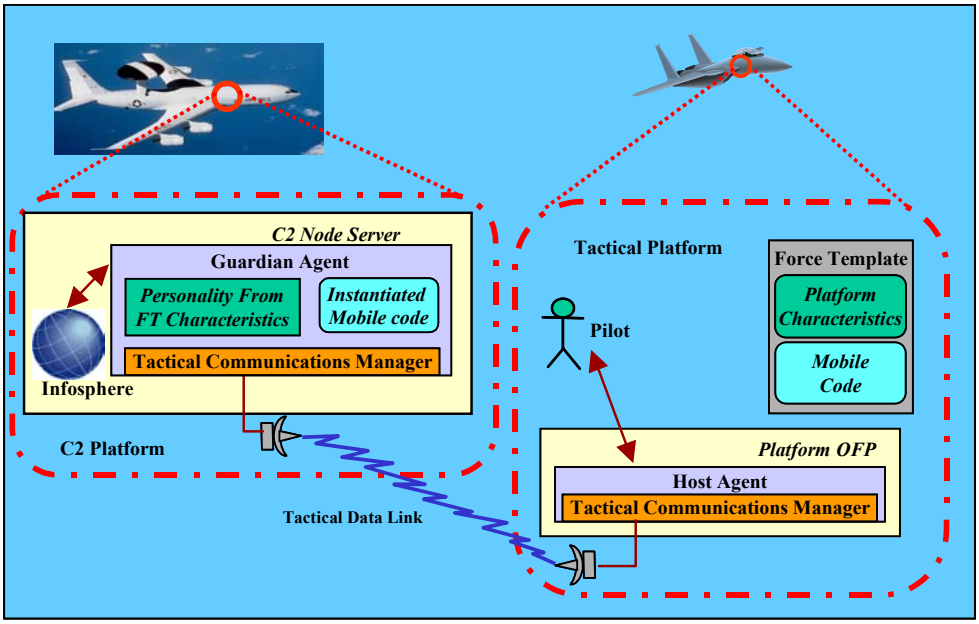

Figure 4 – IEIST Architecture

The Guardian Agent provides the key capability for accessing, filtering, and processing relevant information from the JBI for the platform. It also provides the gateway through which information generated by the legacy platform can be transformed and published to the JBI. For information flowing from the platform, the Guardian Agent provides the bridge to publish platform data to the Infosphere. Such information could include mission updates, weapon status and other platform state data or state changes. The Guardian Agent performs such tasks as evaluating sensor reports for potential threats to its platform. The Guardian Agent "knows" the mission of its platform including the mission route. For each sensor report it receives, it makes an evaluation based on the route and signature of the platform. If the Guardian Agent discovers that a sensor report shows a threat to its guarded platform, it informs the pilot or operator of the platform about this threat. Included in this alert is a recommended route around the threat. An example of another task that the Guardian Agent carries out is its support for dynamic mission

changes such as that required to carry out an attack on newly discovered Time Critical Target (TCT.) The Guardian Agent can plan the proper mission elements for the platform to carry out this new task.

The Guardian Agent is written in Java. The use of Java supports the concept of mobile code. The Guardian Agent has the ability to have custom behavior code for a platform upon instantiation from the Guardian Agent Factory. For maximum flexibility, the threat analysis, re-routing, and automated target/weapon pairing functionality are treated as services that the Guardian Agent uses to carry out its mission.

Guardian Agent Requirements Development

Guardian Agent requirements have been developed using the Unified Modeling Language (UML). The Use Case method was selected as the formal method for developing these requirements. The specific Use Case notation used in the Guardian Agent requirements analysis is documented in tutorials [7,8].

To specify a Use Case, two diagrams are used. These diagrams are the basic Use Case diagram and a sequence diagram. To illustrate what these diagrams look like, Figure 5 and Figure 6 show the basic Use Case diagram and sequence diagram for the Guardian Agent scenario *Threat Analysis with no Threat Found*.

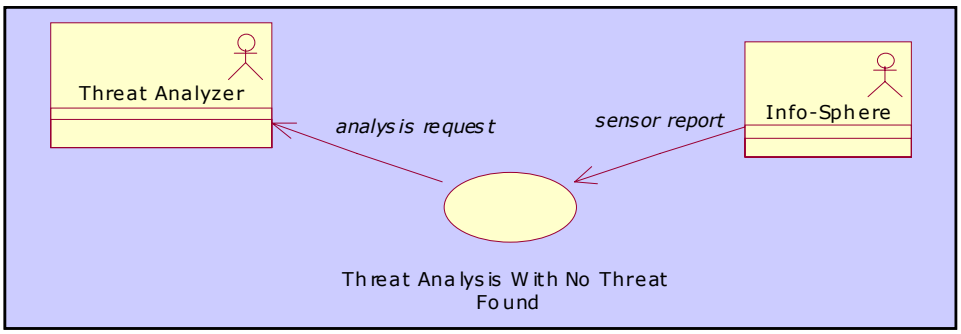

Figure 5 – Basic Use Case Diagram Example

Use Cases were derived from consideration of the global requirements for connecting a legacy system to the JBI [1,6]. These requirements included presenting the legacy platform as a "smart resource" for the Command and Control node. The desired functional requirements for the Guardian Agent can be grouped into the following areas:

- Initialization
- Threat Analysis
- New Mission Support
- Fault Tolerance
- Data And Image Publishing
- Data And Image Query Support
- Sensor Request

The following is a detailed description of the Guardian Agent functional areas for which the formal Use Case analysis was carried out:

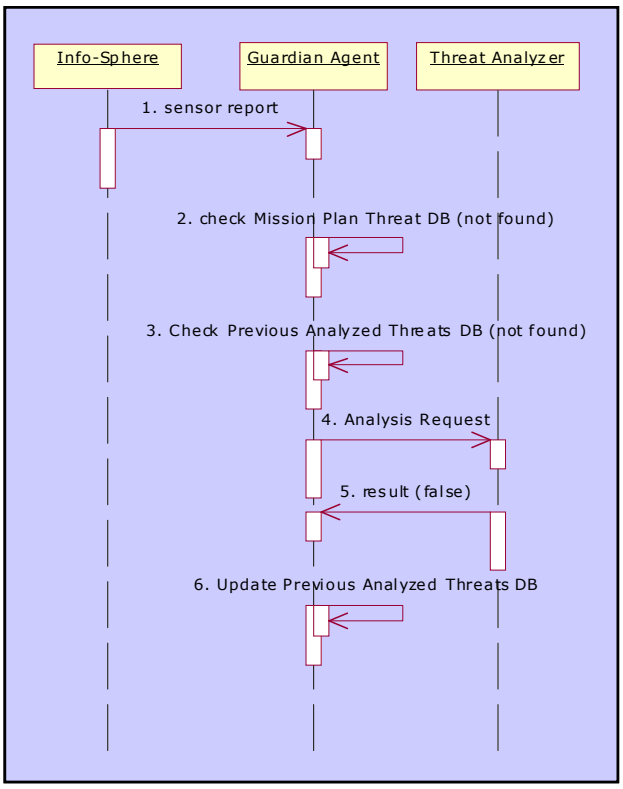

Figure 6 – Sequence Diagram Example

Initialization – The initialization Use Case covers interactions of the Guardian Agent when it is started. A major portion of this analysis specifies how the newly started Guardian Agent connects with the JBI and the Host Agent on the platform.

Threat Analysis – The Use Cases in this area cover the capability that the Guardian Agent provides for intelligently filtering and analyzing sensor reports for potential threats to its guarded platform. The outcome of the threat analysis can be either (1) the potential threat is found to not be a threat or (2) the potential threat is an actual threat to the platform. If an actual threat is found, a new route that avoids the threat while still allowing the continuation of the mission is generated. The pilot or operator of the platform is then informed of the threat and given the proposed new route.

New Mission Support – The capability for a commander to respond to dynamic changes in the battlefield is a critical requirement for the JBI and Guardian Agent-connected legacy platform. The New Mission Support Use Cases specify the functionality that the Guardian Agent must provide to support dynamic mission changes for its platform. This support includes preparing the necessary route and targeting information for the pilot or operator. An example of a dynamic mission change would be the re-routing and assignment of a platform to attack a TCT.

Fault Tolerance – The Use Cases in this area define the interactions and behavior of the Guardian Agent for the case of a fault. A typical fault would include the scenario of the loss of the communications link between the Guardian Agent and the platform. The Fault Tolerance Use Cases describe the Guardian Agent behavior for the occurrence of this fault and the recovery from this same fault.

Data And Image Publishing – The Guardian Agent brings the capability of publishing sensor, image, or other legacy platform generated data to the JBI. The Guardian Agent handles the task of specifying to the JBI what information the platform will publish.

When information such as data or an image is generated by the platform, the Guardian Agent processes and publishes these data to the JBI.

Data And Image Query – One of the objectives of the Guardian Agent is to make the legacy platform appear as a "smart resource" that is available to other clients on the JBI (such as a C^2 node commander or operator.) One way to support this concept is to implement the ability to evaluate and answer data/image queries from the JBI. These queries will be requests for data (such as mission related parameters, weapons status, and others) or for things such as stored image or sensor data generated by the platform.

Sensor Request – One of the objectives of IEIST is to make available the legacy platform as a virtual extension of a global network of sensors and information providers. The Sensor Report Use Case covers the scenario where a client of the JBI makes a request for sensor coverage by some platform flying in a geographical region. An example of this type of request would be that an aircraft provide an image of some region along its current mission route. A constraint on this Use Case is that the sensor request does not have any impact on the current mission of the platform. The Guardian Agent provides the mechanism for implementing this capability via evaluation of the request and only forwarding this request to the platform if it meets the "no impact on the mission" criterion.

Guardian Agent Design

After the "black box" behavior of the Guardian Agent was established through Use Cases, the internal design, that provides the required Use Case behaviors, was developed. The UML Collaboration analysis (Reference 9) was used to define and specify the internal structure of the Guardian Agent needed to carry out the Use Cases. The Collaboration Diagram analysis also specified the order of interaction among the internal Guardian Agent components. The Collaboration Diagram analysis for the *Threat Analysis With No Threat Found* example Use Case shown in the previous section is given in Figure 7.

A feature of the Collaboration diagram analysis is that the Use Cases provide the input to this step of the Guardian Agent design. Because the Use Cases provide the analysis input, the system requirements of the Guardian Agent were mapped into a structure that can be implemented in a programming language. Since the Collaboration diagram is a detailed analysis of the internal structure and interactions for a Use Case, it is possible that different structures and interactions within the Guardian Agent can produce the same result as viewed external to the system (the Use Case view.) Therefore, it is possible that there is more than one Collaboration diagram for a given Use Case.

The *Threat Analysis With No Threat Found* Use Case provides an example of where a single Use Case results in more than one Collaboration diagram. For this Use Case, the Guardian Agent receives a sensor report of a potential threat, the threat is analyzed by the full-featured threat analyzer, and the result is that the potential threat is not a threat to the platform. The Collaboration diagram in Figure 7 shows the case for this Use Case where

the threat has been analyzed before and found to not be a threat. However, the threat has moved beyond an acceptable range established for this platform. Therefore, a full re-analysis of the potential threat is required. As in the original case, the potential threat is found to still not be an actual threat to the current platform mission.

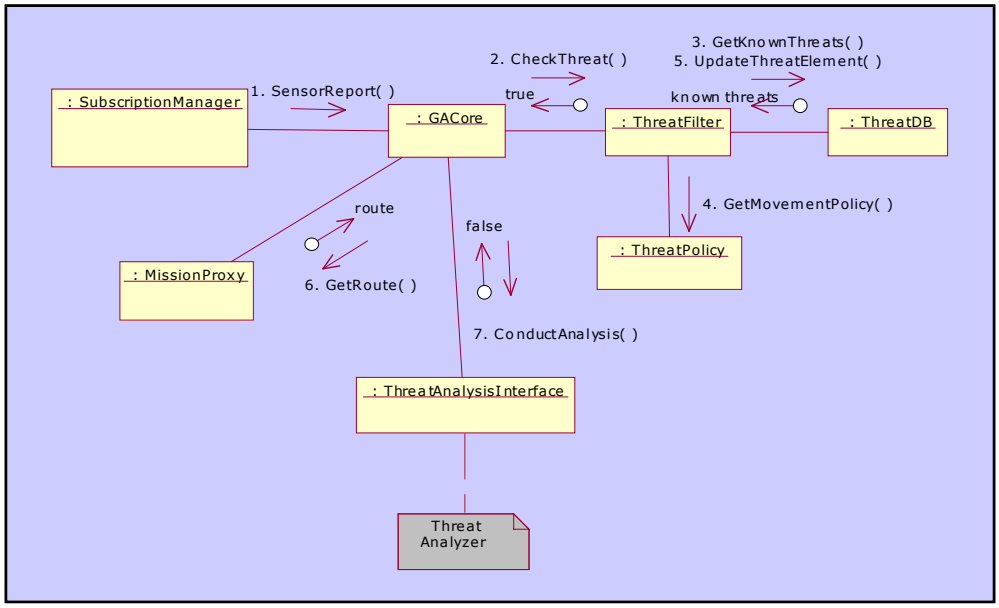

Figure 7 – Collaboration Diagram Example for the Threat Analysis With No Threat Found Use Case – Threat Has Moved.

An alternative Collaboration Diagram sequence of internal actions within the Guardian Agent for this same Use Case is shown in Figure 8. For this Collaboration Diagram, the sensor report is for a potential threat that has not been analyzed by the threat analyzer. The threat analysis by the threat analyzer shows that the potential threat is not an actual threat. From the standpoint of the external entities to the Guardian Agent, the interactions (the inputs and outputs) into and out of the Guardian Agent are the same for both the Collaboration diagrams shown in Figures 7 and 8.

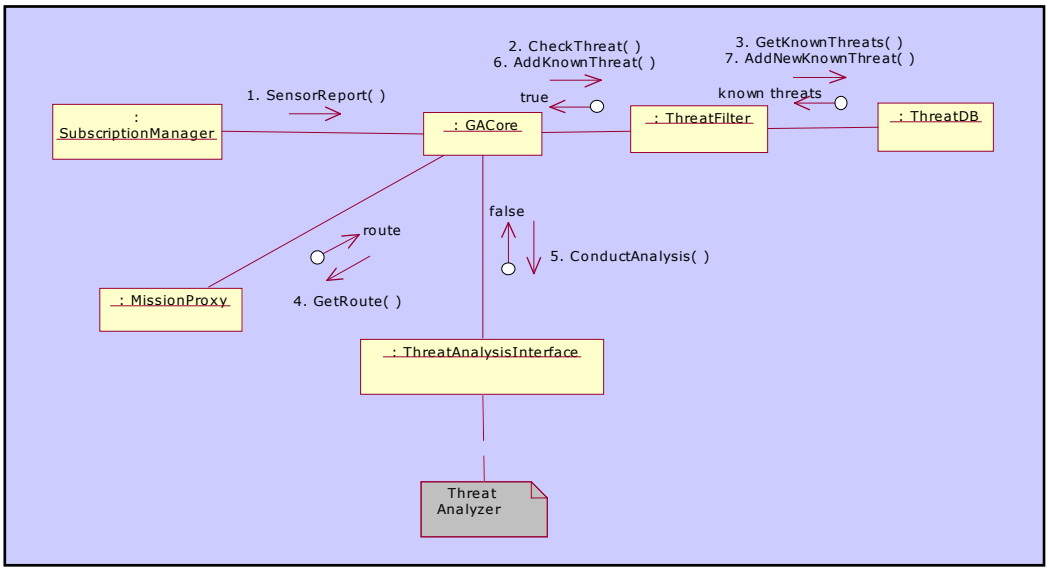

Figure 8 - Collaboration Diagram Example For No Threat Found Use Case – Fixed Threat

However, the Collaboration diagram analysis determines that there are multiple internal behaviors within the Guardian Agent for this Use Case.

Once all the Collaboration analysis diagrams were completed, they were mapped into a single Object-Oriented class diagram for the Guardian Agent. This final diagram provides the classes that can be programmed using a language such as Java.

Guardian Agent Modularity

The Use Case and Collaboration Diagram analysis produced a design for the Guardian Agent that can be directly implemented in the preferred programming language. For the IEIST program, the Guardian Agent design determined by the output of the Use Case/Collaboration Diagram analysis process has been implemented in Java. These UML methods guarantee that the requirements for the Guardian Agent are contained in the final design. The resultant Guardian Agent design is shown in Figure 9.

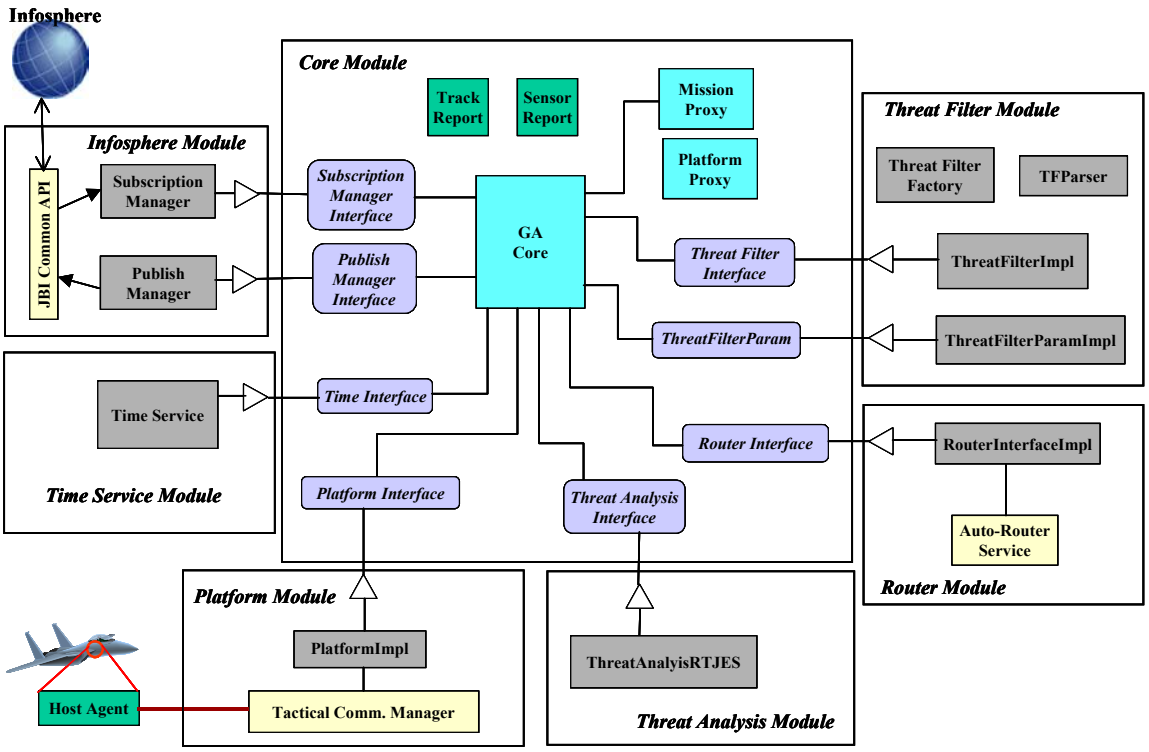

Figure 9 – Guardian Agent Design

One of the implementation requirements is to allow the Guardian Agent to support different types of legacy platforms. The same implementation desire can be extended to the "enterprise services" that the Guardian Agent needs to use to do things like threat analysis, auto-routing, and other tasks. In addition, it is probable that the JBI interfaces will continue to change and evolve after the Guardian Agent is implemented.

To achieve independence of the Guardian Agent from a specific implementation of an enterprise service such as a threat analyzer or to a platform type, the Guardian Agent uses the design approach of a *generic core* and *abstract interfaces*. If you look at the Use Case and Collaboration diagram analysis, the functionality of the Guardian Agent is independent of a particular platform type, service implementation, or Infosphere implementation. This generic functionality can be implemented by set of core classes as shown in Figure 9.

The Guardian Agent core module contains an object called the GACore. This object controls the operation of the Guardian Agent. The Guardian Agent core module also consists of a series of abstract interfaces that can be accessed by the GACore object. These interfaces specify a common method for the core to interact with different implementations of an Infosphere, platform type, threat analysis service, and other services.

The Guardian Agent core module also contains an implementation for the common data types that it needs to manipulate. These data classes include representations of such things as sensor reports, track reports, and other data. The Guardian Agent core also includes a generic representation of the platform's mission (via the mission proxy class) and platform physical properties (via the platform proxy class). The Guardian Agent core is implemented in the Java programming language.

As can be seen in Figure 9, the other portions of the Guardian Agent design consists of a series of "modules" that provide concrete implementations of the abstract interfaces in the core. As an example, the *Threat Analysis Interface* specifies the way the Guardian Agent core interacts with some implementation of a threat analysis service. The Threat Analysis Module provides a concrete implementation of a threat analyzer. The actual implementation can vary due to the type of platform or level of analysis sophistication desired by the military service using the Guardian Agent. With this design approach, different implementations of a threat analyzer can be seamlessly supported by the same Guardian Agent core code.

As an example, a manned platform may use a threat analyzer that takes into account a very detailed 3-D signature. While a cheap UAV platform may only require the use of a much simpler 1-D signature analysis. The Guardian Agent can accommodate both threat analysis implementations through the use of modular design and abstract interfaces. The differences in the implementations are isolated to the Threat Analysis module. The Guardian Agent core module code remains unchanged. This same principle of design applies to the other implementations for the Infosphere, platform, router, and threat filter modules.

Benefits to the C^2ISR

The benefits of Network Centric Warfare and the GIG to the C^2ISR community are well documented, and beginning to be understood. Commanders are better able to coordinate

their war fighting resources with their latest intelligence. In theory, the flow of information from the commander to and from his war-fighters is greatly improved with this new infrastructure. But the connectivity to the war-fighter still needs considerable improvements which allow him to publish and subscribe to the C^2 resources.

Guardian Agents and Force Templates are technologies that greatly improve this war-fighter connectivity. The Guardian Agent reviews, filters, and formats information that is pertinent to the war-fighter. It also is able to publish time critical information that the warfighter discovers to the C^2. The Force Template provides a way for legacy war-fighting systems to be understood and enhanced so that they can play in the Network Centric engagement.

Summary/Conclusions

This paper has addressed the problem of improving the kill chain of Time Critical Targets by better equipping legacy war-fighting platforms with access to the Global Infosphere Grid. This investment in technology is in step with the rapid shift of commanders from platform centric warfare to network centric warfare. This shift moves from overwhelming numbers and power to overwhelming intelligence, inter-operability, and response.

The Joint Battlespace Infosphere is the Air Force approach to managing information in this new vision. The JBI offers key services for moving information, but also adds value by monitoring the information, providing information persistence, and evaluation of the information as it is operationally used.

The key objective of this paper was to formally introduce the Guardian Agent Technology. Specific details are presented which show the design, creation, instantiation, and use of Guardian Agents. Detail was also given to other components such as the Force Template and the Tactical Communication Manager, and Host Agent, which provide the inputs and environment for the Guardian Agent.

Once instantiated (or enabled) on a C^2 Platform, a Guardian Agent can affectively add considerable capability to a war-fighting platform by delivering decision quality information into the cockpit in a timely manner. In reverse, the war-fighting platform can also deliver time critical information it discovers back to the C^2 and ultimately, the Commander.

Enabling the legacy platforms as information users and suppliers ultimately completes the requirements of connectivity between elements of the Global Information Grid.

References

1. For AFRL/IFTA by Boeing Phantom Works, *Incremental Upgrade of Legacy Systems for Common Battle Management System Battle-Management Elements (IULS-CBE) Study Program Final Report*, 25 January 2002.

2. [USAF, 1999] *United States Air Force Scientific Advisory Board Report on "Building the Joint Battlespace Infosphere"*, Volume 1: Summary, SAB-TR-99-02, December 17, 1999.

3. [USAF, 1999] *United States Air Force Aerospace Command Control Intelligence, Reconnaissance (C^2ISR) Campaign Plan 2000*, December 23, 1999.

4. [USAF, 1997] *Chairman of the Joint Chiefs of Staff, "Joint Vision 2010",* May, 1997.

5. [USAF, 2000] *Chairman of the Joint Chiefs of Staff, "Joint Vision 2020",* June, 2000.

6. Satterthwaite, C. P., Corman, D. E., and Herm, *T. S., Transforming Legacy Systems To Obtain Information Superiority,* 6th International Command and Control Research and Technology Symposium, U. S. Naval Academy, Annapolis, MD., June 2001.

7. *The Unified Modeling Language User Guide*, Grady Booch, James Rumbaugh, and Ivar Jacobson, Addison-Wesley, 1999.

8. UML Use Case Tutorial Document.
http://www.objectmentor.com/resources/articles/usecases.pdf

9. *OMB Unified Modeling Language Specification*, v1.4, Object Management Group, September 2001. Reference located at http://www.uml.org

10. Corman, D. E., Gossett, J., "*WSOA – Using Emerging OSA Standards to Enable Innovative Techniques for TCT Prosecution*", 20th DASC, IEEE/AIAA, October 2001.

11. *Power to the Edge, Command Control in the Information Age,* Alberts, D. S., Hayes, R. E., Center for Advanced Concepts and Technology, June 2003.

12. *Network Centric Warfare, Developing and Leveraging Information Superiority,* Alberts, D. S., Garstka, J. J., Stein, F. P., R. E., Center for Advanced Concepts and Technology, August 1999.

13. AFRL/IF Joint Battlespace Infosphere Office, Force Templates: Standardized Client Information Interface Descriptions for the Joint Battlespace Infosphere, JBI Concept Definition Document, Version 4.0, July 2003.