Building an Experimental Joint Battlespace Infosphere (YJBI-CB)

Douglas Holzhauer[1], Vaughn Combs, Mark Linderman, Robert Duncomb, Jason Quigley, Mark Dyson, Robert Paragi, David A.Young
Air Force Research Laboratory/ Information Directorate, Rome, NY

Digendra Das, SUNY Institute of Technology at Utica/Rome, NY

Carrie Kindler, ITT Industries, Rome, NY

Abstract

Targeting can be more complicated than just locating a facility or military column. If the target is suspected of having chemical weapons, weather forecasters and armament experts need to work cooperatively to decide how to disable the target without creating chemical fallout that endangers non-combatants or friendly forces. However, this collaboration is complicated by the fact that Air Operation Center's (AOC) functional units are often spatially distributed and have "hard-wired" communication lines. Additionally, previous combat situations have shown that missions may change during the conflict. This condition may necessitate a change in information dissemination patterns to route information to new users that previously did not require it. The Joint Battlespace Infosphere (JBI) provides the flexible conduit to dynamically handle these changes in information needs.

AFRL/IF is evaluating JBI technology through an in-house and contractual prototype program, which develops and tests JBI instantiations. This JBI instantiation emulates four AOC functional cells; the weather cell, the bio-environmental/civil engineering cell, the targeting cell, and the Joint Force Air Campaign Commander (JFACC); and two components of the Theater Air Control System; the sensor network and the sensor fusion facility. These cells currently use stand-alone planning and analysis software codes including; the MM5 weather model, Joint Targeting Toolbox, Hazard Prediction and Assessment Capability (HPAC), and the FUSION TRACKER. This demonstration expands these stand-alone systems into a virtual network, that uses subscribe and publish capabilities to share information between the application programs. The ability to subscribe to only the published information of interest empowers warfighters to make in-field information flow changes, thereby assuring that the right information is delivered to the correct user in a timely manner. To demonstrate this technology, a targeting and battlespace awareness scenario is used. The scenario involves assessing the effects of possible chemical/biological collateral damage based on the attack of a facility that manufactures chemical weapons. This JBI experiment is a critical step towards total integration of the latest information technology into a command and control system.

1. Introduction

[1] Correspondence to: Dr. D. Holzhauer, Air Force Research Laboratory/IFTC, 26 Electronics Parkway, Rome NY 13441-4514

The Joint Battle Infosphere (JBI) is a combat information management system that provides individual users with the specific information needed to accomplish their functional responsibilities during a crisis or conflict. The JBI integrates data from a wide variety of sources, aggregates this information, and distributes the information in the appropriate form and level of detail to users at all echelons. [2] With this goal, the Air Force Research Laboratory's Information Directorate initiated a research program that consists of three major focus areas; development of JBI middleware (middleware is the connection software), integration of legacy command and control clients, and development of enabling science and technology. As a part of these focus areas the Information Directorate is developing JBI prototypes (Y-JBI). These are termed "Y" JBIs since they are analogous to the "Y" prototype fighter aircraft, as YF-22. Y-JBIs are not entire system prototypes, but are meant to identify and evaluate specific JBI technologies for viability. The Y-JBI prototypes emphasize either the middleware technologies, or the client integration challenges. The YJBI-CB was developed to evaluate the viability of the AFRL subscribe and publish packages to support the information sharing needs of a subset of Air Operations Center Cells. The emphasis is on the subscription and publication of numerically intensive objects.

JBI technology is demonstrated by emulating several functions of an Air Operations Center (AOC). Applying this technology to an actual Air Force system allowed the YJBI to publish and subscribe real world information and enabled the technology to be demonstrated in a form that is meaningful to operational units. AOC functional cells are emulated with the goal of demonstrating information sharing between units, not an integration of the application computer programs. Some clients are publishers, some are subscribers and some are both. These cells use state of the art software to perform their functions, which runs in a stand-alone mode typical of legacy software.

The scenario we address is that of evaluating the effects that an air strike on a chemical manufacturing facility would have on the local population. Predicting where the contaminants will go after their initial release is largely dependent on the weather conditions at the time of the incident, with the elements of greatest interest being the wind direction and speed. The solution requires cooperation between three separate cells of the AOC, with final reporting to the Commander.

2. The AFRL Publish and Subscribe Package

The publisher package comprises a collection of Application Program Interface (API) method signatures for a PubAdapter class and other supporting classes. The PubAdapter class provides methods that allow a publishing application to register, publish, and unregister. In addition, a number of methods are provided to monitor the state of the publisher (e.g. the current number of matched subscribers). As part of the registration process, a publishing application specifies invariant data describing the information objects to be published. This information will be used to support the brokering process. A PubAdapter may simultaneously publish several information object types – each is registered and managed independently. Analogously, the subscriber package comprises a collection of API method signatures for a SubAdapter class and other supporting classes. The SubAdapter class provides methods that allow a subscribing application to register, receive published information objects, and unregister. Similar to the PubAdapter’s invariant

[2] Report on Building the Joint Battlespace Infosphere, SAB-TR-99-02, December 17, 1999

metadata, the SubAdapter allows a subscriber to specify a template to support the brokering process. Currently the underlying template system permits equality matching with wildcarding with respect to the supplied metadata template attribute values. Currently a subscription may contain an arbitrary XQL or XPATH expression that is used to perform content-based filtering of published documents. The filtering is done within the PubAdapter support classes and is entirely transparent to the publishing application. This feature could be used to significantly reduce the amount of data that is actually disseminated to registered subscribers. Additionally, the subscribing application may specify an indirect recipient as part of the subscription. If this is the case the PubAdapter will forward (via email) a copy of the document to the specified recipients. The AFRL Publish and Subscribe package makes use of Sun's JINI™ Network Technology as an underlying middleware layer. (Figure 1) After the package is installed the users of the package APIs need never know (nor care) about JINI or any of its classes or interfaces.

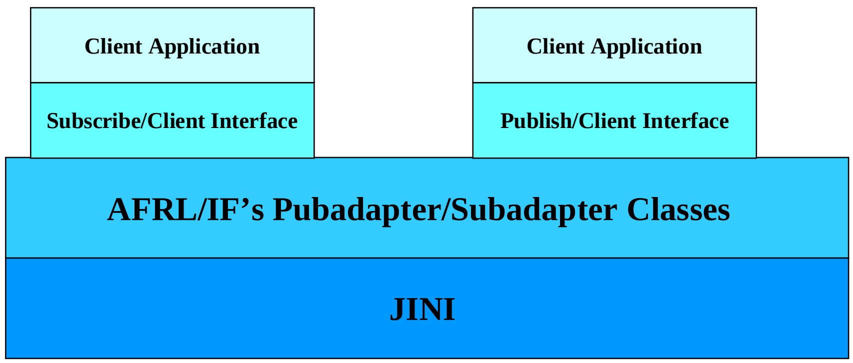

Figure 1 - Software Classes

3. YJBI-CB System Design

One major concept of the JBI is that of information exchange through "publish and subscribe". An additional desired capability is for clients to find information objects in a repository by matching a search pattern; a query capability. The information exchanged between clients is stored in Java object form. This exchange is a two step process using two associated objects, a metadata object and a published object. The metadata object describes the structure and the meaning of the published object's represented information and is used for matching publisher and subscriber clients. Each published object contains both the represented information, referred to as the payload, and metadata (information about the payload). YJBI-CB defines the published objects as ClientObjects. The metadata is encapsulated in the header variable in XML format and the payload is stored as attributes and values.

YJBI-CB defines only two metadata objects: ClientMetadescriptor and RegionMetadescriptor. These are essentially the same except that the RegionMetadescriptor has been augmented with an additional field "Region", that defines location for weather information. The fields used are:

- Publisher - Name of the AOC cell that is publishing this information
- Name - Identifies the ClientObject
- Type - Type of data of the encapsulated information (text, gridded, image)

- DocUID - Unique document identifier
- Encoding - File extension of the payload (XML, PRF, …)
- DeliveryMechanism - Object delivery mechanism (is it encapsulated in the ClientObject, or do we have to do an HTTP transfer)
- Region - Geographical location of the information

An example of the values associated with these fields is given using the cloud information. There are five different objects that have the name "cloudcover".

Name	cloudcover	cloudcover	cloudcover	cloudcover	cloudcover
Type	image	image	image	gridded	gridded
Encoding	gif	gif	gif	xml	xml
Delivery	object	object	object	object	object
Region	northeast	southwest	usa	northeast	southwest

These five cloudcovers indicate (in order) satellite images of the Northeast, Southwest and the entire United States, stored in a .gif format, and numerical data, stored in an x-y-z grid format, for the Northeast and the Southwest. A match of all the metadata fields must occur to define the specific payload information subscribed to by a JBI client.

For most YJBI-CB clients there is legacy software currently used as a stand-alone package. When these are integrated into a system, the clients can either automatically run the legacy code or continue with a person in the loop. For YJBI-CB, only the FUSION TRACKER is automated. All other clients transfer the information object to the subscribing AOC cell, then cast the object into a file of the correct format and which is then used by the legacy code independently from the YJBI-CB client.

4. YJBI-CB Clients

Six functional cells from a Theater Air Control System (TACS) form the YJBI-CB: Bio-Environmental Cell, Weather Cell, Targeting Cell, Control & Reporting Element, Control & Reporting Center, and the JFACC. The TACS functional diagram for these cells is shown in Figure 2. Each cell has its respective application code listed below the cell title.

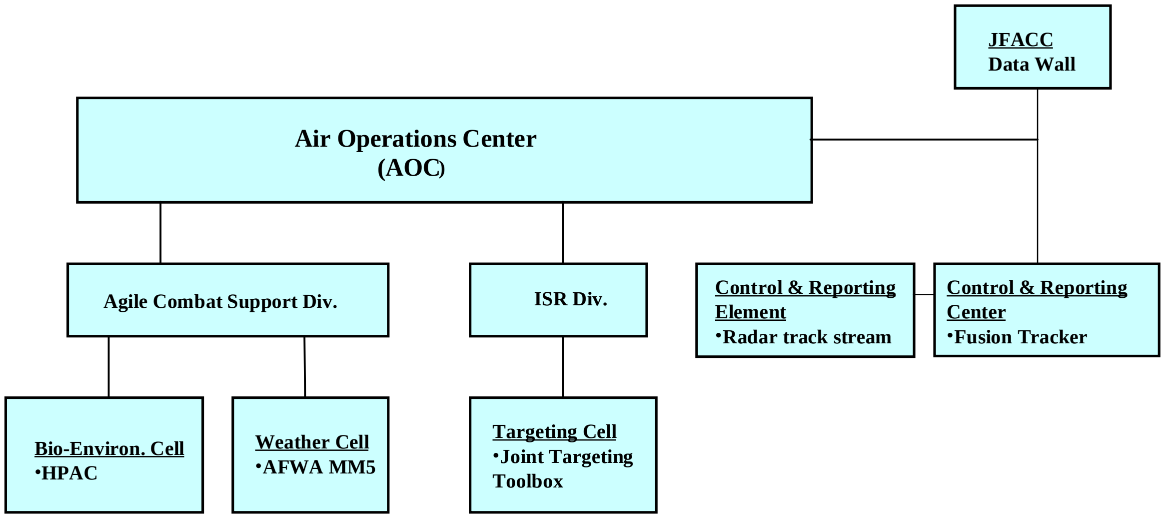

Figure 2 - TACS Functional Cells Emulated

Weather Cell: An automated file transfer protocol (ftp) script brings the latest numerical MM5 (45 km) data to a local database. This GRIB data is then post-processed to index each forecast element in the database by time and height. A script defining the location of the incident then extracts a virtual vertical wind profile for the point of interest or a set of wind profiles for a (user defined) area of interest around the chemical release site. This profile data is encapsulated in a ClientObject and published. The wind profile information is used by the Hazard Prediction and Assessment Capability (HPAC) computer program, which produces the plume forecast. This method increased the timeliness of response by hosting the MM5 data locally, which decreases communications and bandwidth dependencies and allows multiple runs of the same scenario with varying inputs to factor in uncertainty.

Bio-Environmental Cell: This cell is both a subscriber and publisher of JBI objects. The application code associated with this cell is the (HPAC) code[3]. This code models the transport of chemical and biological materials based on the terrain, wind vector profiles and incident parameters. Thus the HPAC model predicts the geographical area that will be contaminated by the chemical materials over the next 24 hours. This capability can function as a stand-alone code but is being used in the YJBI to show cooperation and information sharing between the Weather cell and the Bio-environmental cell.

Targeting Cell: Targeting a chemical facility is the major mission thread in the YJBI-CB. The target location and time of strike are published as an information object from this cell.

Control and Reporting Element: We have added an additional mission thread that makes use of the high performance computing facility at AFRL/IF. In this ancillary thread, the Control and Reporting Element publishes a time sequenced file that contain radar tracking data from several sensor locations.

[3] HPAC is a product of the Defense Threat Reduction Agency

Control and Reporting Center: The Control and Reporting Center uses a FUSION TRACKER program to reduce the multiple tracks that result from multiple sensors to a consistent set of target tracks that are useful for command and control decisions. This client uses the distributed processing capability of the AFRL/IF cluster computer (Figure 3) to accomplish its fusion. This demonstrates the interfacing of the AFRL High Performance Computing Facility to the Command and Control center represented by the Data Wall.

Figure 3 - FUSION TRACKER runs on AFRL/IF's Cluster Computer

JFACC: The JFACC client is the situation commander's application in our scenario. The situation commander is the person (or group) that is the final receiver of the information in its highest fused state. As with a pilot in a fighter aircraft we do not want information overload, but want to present the commander with decision quality information. YJBI-CB uses the AFRL/IF developed Web Enabled Timeline Analysis System graphics code (WebTAS) to present the battle picture to the commander. The JFACC is actually two software packages that reside on one computer and since they both are written in JAVA, can access the classes and methods from each other. This makes for a seamless integration. The JFACC-GUI is the code that allows the commander to subscribe to the necessary JBI information objects that are then displayed on the Data Wall (Figure 4) using WebTAS. WebTAS supports the superpositioning of the information onto the Data Wall. The Data Wall interface first layers a National Imaging and Mapping Agency (NIMA) digital map of the combat area, then layers the digital cloud and chemical plume data at the correct latitude and longitude locations. This superpositioning is very important for rapid battlefield awareness.

The Data Wall is an enhanced computer display that tiles the output of multiple computer displays into a single workspace with a display resolution of approximately 5.8 million pixels and covers a 12' x 3' screen area. In addition to the vast increase in display area, it also enhances human and computer interaction by providing user control of the computer through an independent speech recognition system and a camera-tracked laser pointer system.

Figure 4 - AFRL Data Wall

5. Demonstration Scenario

To demonstrate the YJBI-CB an example incident of a planned attack of a chemical manufacturing facility is used. To help track the publishing and subscribing of the ClientObjects on the YJBI, a time flow diagram (Figure 5) is furnished. The chemical facility is identified as a possible target by the Targeting Cell, which then publishes a Target Nomination List (TNL) object. All other AOC clients require the TNL information and therefore subscribe to receive it. At each cell, the Target Nomination List pops up in a JAVA frame, which is then read by the users to determine their necessary response. If the Bio-Environmental Cell notes a weapon of mass destruction (WMD) facility on the TNL, a subscription to the Profile data is made to obtain the wind vector profile information needed by the HPAC program. The JFACC Cell subscribes to the TNL, the cloud image, the cloud gridded data, and the predicted chemical plume dispersion.

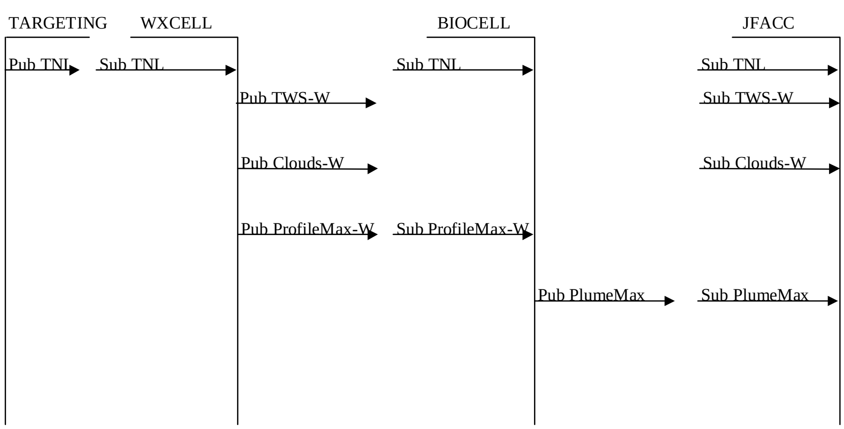

Figure 5 - Subscribe and Publish Flow

Then the Weather Cell and the Bio-Environmental Cell perform their collaborative analyses in support of the mission. The Weather Cell obtains the current theater weather satellite (TWS) images for the location associated with the mission, and obtains the wind velocity data cube (ProfileMax, a one degree latitude by one degree longitude wind velocity box centered on the chemical factory). This Weather Cell publishes the velocity data cube (ProfileMax object), several satellite cloud images (TWS object) and the gridded cloud objects. The Bio-Environmental Cell performs the HPAC analysis upon receipt of the profile data and, in turn, publishes the plume object. Using the Bio-Environmental Cell as an example, we can see that the subscribing and publishing is done in a two step process.

In Figure 6, the TNL and Weather metadata used for subscribing are shown. First these are selected by highlighting them, after which the register button is clicked, resulting in the registering of the Bio-Environmental Cell subscription. The actual subscription is made automatically at this point. For the publishing the two steps, registration and publication, are done explicitly with the registration button clicked, then the publish button being clicked. As with the subscription, the Plume metadata (either the gridded or the gif image) must be selected by highlighting before publication occurs.

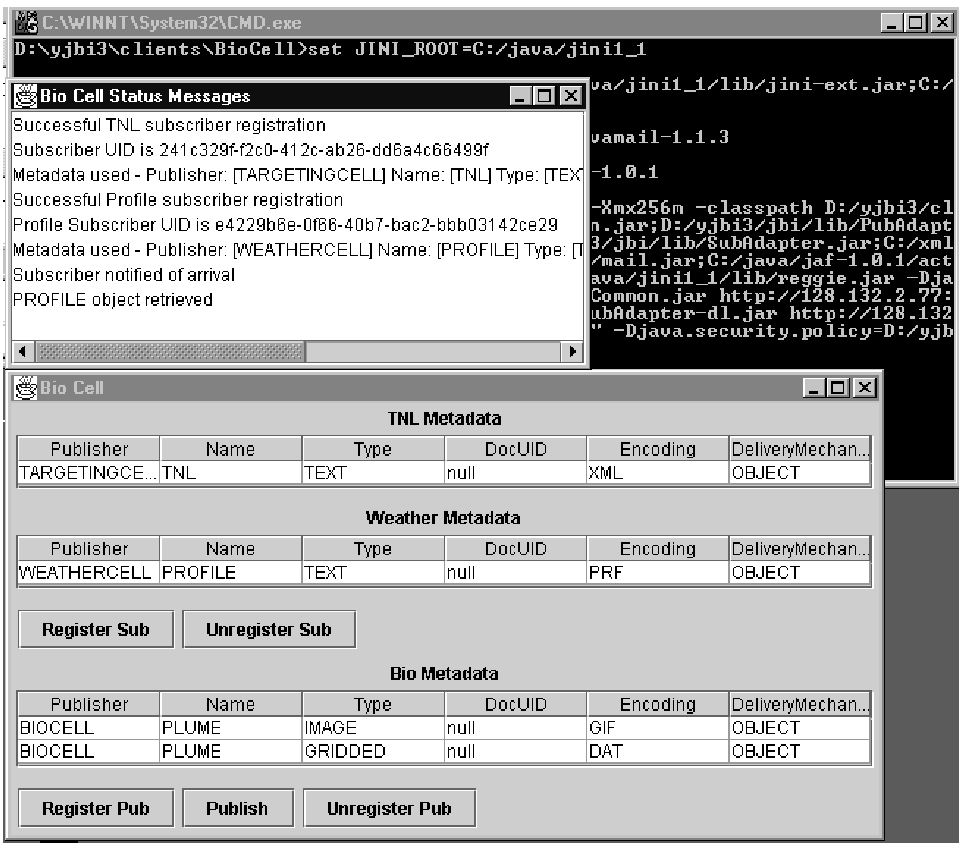

Figure 6 - Screen Capture of BIO-CELL GUI

In the YJBI-CB, the JFACC Client is the receiver of decision quality information for the commander. The JFACC Client presents the information on the Data Wall overlaying the clouds or the plume on the NIMA map. The plume overlay is shown in Figure 7.

This demonstration is accomplished using four separate computers connected by the Rome Research Site LAN. Each computer has JAVA and JINI installed as well as the software specific to its function. As such, it is non-intrusive and can be added to any network.

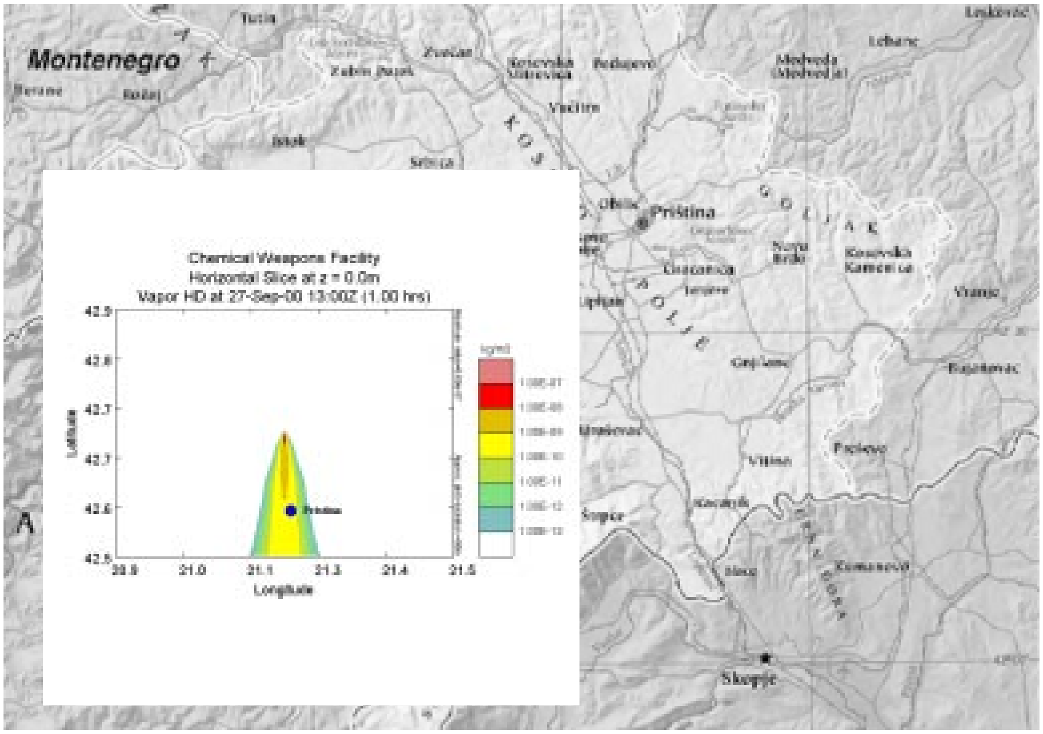

Figure 7 - Dispersion of a chemical agent shown on JFACC computer

6. Conclusions

This very early demonstration of a Joint Battlespace Infosphere has shown the interaction of several clients using the AFRL publish and subscribe packages. Additional research areas include publishing large information files, metadata ontology, JBI API standardization and addition of a query capability (and its associated repository).

7. Acknowledgements

We thank the following for their assistance in this project. Defense Threat Reduction Agency (DRA) provided HPAC and technical assistance with the GRIB to HPAC conversion code. Dr. Charles Pederson (AFRL) provided the FUSION TRACKER Code and assisted in the development of the script to run it. Mr. John Mucks provided the WebTAS code. Air Force Weather Agency provided the GRIB MM5 data needed to make this all work.