The Outlook Y-JBI: An E-Mail Based Approach to Prototyping the Joint Battlespace Infosphere

Major Robert E. Marmelstein, Ph.D.

Air Force Research Lab (AFRL) Information Directorate,
Information Systems Division (IFS)
525 Brooks Rd
Rome, NY 13441

Abstract

One approach for rapidly implementing a prototype Joint Battlespace Infosphere (Y-JBI) publish and subscribe (P&S) system is to leverage the existing e-mail transport infrastructure. E-mail has the advantage of being standardized, pervasive, robust, and scalable. This Y-JBI is implemented though the peer-to-peer interaction of publisher and subscriber agents. These agent pass subscription requests, acknowledgements, and information objects to each other using Extensible Markup Language (XML) e-mail messages. Under this approach, Microsoft Exchange® is employed as the enterprise mail server with Microsoft Outlook® as the primary e-mail application servicing the JBI user. This system enables a direct flow of information from the publisher to subscriber-side fuselets that interface to Microsoft Office® planning tools.

1. Introduction

This paper describes an e-mail based P&S system built as part of an AFRL initiative to evaluate approaches for implementing the JBI concept. This prototype JBI (called the Outlook Y-JBI) leverages widely available Commercial Off the Shelf (COTS) and World Wide Web committee (W3C) standards, such as Extensible Markup Language (XML) and XML Style Language (XSL). We show how the system can be utilized by warfighters to identify, tap into and exploit diverse and distributed information sources that might not otherwise be available. As a result, integrating a mature version of this system into the joint Command and Control (C^2) infrastructure would improve situational awareness and streamline process automation at all levels of the command structure.

2. Background

In the last decade it has become increasingly apparent that information is the key to victory in modern warfare. This is illustrated by the emphasis that Joint Vision 2020 **[4]** places on information (and decision) superiority as an enabler to achieve full spectrum dominance. In order to transform this vision into reality, we must recognize that *getting the right information to the right user at the right time* is the fundamental challenge for military C^2 systems.

Meeting this challenge will not be easy. The DoD has a diverse variety of dedicated C^2 systems, which are frequently optimized to accomplish a single task. Unfortunately, many of these stove-piped systems cannot readily share information with each other. While C^2 architectures such as the Theater Battle Management Core Systems (TBMCS) provide a framework for sharing information among mission applications, databases and the interfaces between applications which cannot be changed to meet dynamically changing information needs. Due to the blizzard of information contained in modern command centers, warfighters find it difficult to obtain the information they need to accomplish the mission efficiently. Often, the information is buried in an application specific database, with no good way to find, extract, and display the needed information. To resolve this state of affairs, the warfighter needs ways to:

- Find needed information
- Access (subscribe to) that information in a flexible manner
- Transport the information from its source to the warfighter.
- Aggregate and visualize the subscribed to information objects.
- Detect critical events by mining multiple information streams; generate real-time alerts in response to these events.
- Push information to other applications.
- Get information to their particular information appliance, be it a workstation, personal digital assistant (PDA), or cellular phone.

3. The Joint Battlespace Infosphere

The Air Force Scientific Advisory Board (SAB) studied the above problem over a period of two years. The resulting reports, Information Management to Support the Warrior (1998) **[5]** and Building the Joint Battlespace Infosphere (1999) **[6]** chart the way ahead for re-inventing the C2 information management. Key definitions derived from the SAB report that describe the Outlook Y-JBI are given in Table 1. The SAB vision for the JBI encompasses the four key concepts described below.

Information exchange through publish and subscribe. This capability enables the user to locate and subscribe to information resources available within the JBI. Each publisher is responsible for tracking users that have subscribed to its resources. When an information resource is published, a tailored version of that resource is forwarded to the subscriber.

Transforming data to knowledge via fuselets. Fuselets are small programs or scripts that process incoming information objects received from established subscriptions. When these objects arrive, fuselets can then aggregate, correlate, and/or transform them into information of interest to a given subscriber. For example, E*Trade® advertises a service that alerts users via e-mail when a given stock meets a specified price target. From the military perspective, a commander might subscribe to intelligence reports for surface to air missile (SAM) sites and the target nomination list (TNL) for the current air tasking order (ATO) cycle. Upon receipt of these resources, the fuselet would compare them to see if they satisfy some predefined condition (e.g., an intelligence report says that a SAM site on the TNL has moved) and bring that fact to the attention of the user (via an alert). This alert can, in turn, be published for the benefit of other users—in this way, fuselets function as both consumers and producers of information.

Distributed collaboration through shared, updateable knowledge objects. This concept refers to the ability of the JBI to facilitate collaborative problem solving among multiple, diverse users. For example, a number of different users might interact to develop a plan for a given mission or task. In this context, the logistician's contribution to the plan might take quite a different form than that of a squadron commander. Despite this, JBI would provide tools enabling each participant to view/modify the plan in a way that is tailored for their specific role while maintaining the consistency and integrity of the unified plan.

Assigned unit incorporation via force templates. The force template is a software description of a military unit that enables that unit to 'plug into' the JBI. Since there is no one JBI system, the actual content of a force template is still to be determined. However, it is likely that force templates will include information about the particular unit (e.g., personnel, command structure, communication infrastructure, equipment inventories, etc). From an information exchange perspective, however, the force template will provide the unit's projected information resources (what it intends to publish) and requirements (information it needs to subscribe to in order to accomplish its function).

Table 1 - Basic Y-JBI Definitions

Publisher	Any entity that promises to make information resources available to other Y-JBI users. The users access these resources via a subscription.
Subscriber	Any user that requests information objects from a Y-JBI publisher
Component	Any entity (publisher or subscriber) connected to the Y-JBI.
Information Resources	A type of resource (document, web page, database, or application) that is made available by a publisher to potential subscribers.
Subscription	A request for information from a publisher from another Y-JBI user that is accepted and acknowledged by the publisher.
Information Objects	Information generated by an information resource in response to a particular subscription. The content of the subscription defines how a subscriber wants to access a given information resource and provides a blueprint for generating an information objects. The resulting object is returned to the subscriber.
Fuselets	Simple programs that execute on behalf of a given subscriber. Fuselets serve to process (aggregate, correlate, and transform) incoming information objects into a form desired by the user.

4. Outlook Y-JBI Architecture – Spiral 1

One promising approach for rapidly implementing a JBI-like P&S system is to leverage the existing e-mail transport infrastructure. E-mail has the advantage of being standardized, pervasive, robust, and scalable within the DoD. The Mail Application Programming Interface (API) Messaging Benchmark is frequently used to test e-mail server performance under a simulated user load **[1]**. Figure 1 shows that, on average, state-of-the-art enterprise e-mail servers can serve tens of thousands of simulated users generating millions of e-mails per day while maintaining response times of less than 1 second. These results illustrate the promise of using e-mail to support high volume P&S traffic within a command center. Despite the impressive performance of individual mail servers, real-time mail delivery (e.g., less than 5 seconds) cannot be guaranteed due to external network latencies. As a result, we view the Outlook Y-JBI as a solution for delivering information objects in near real-time.

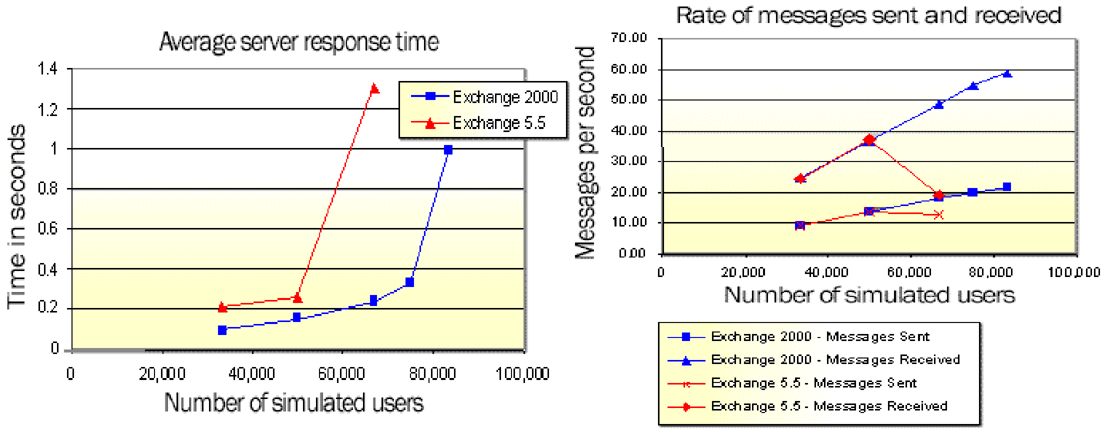
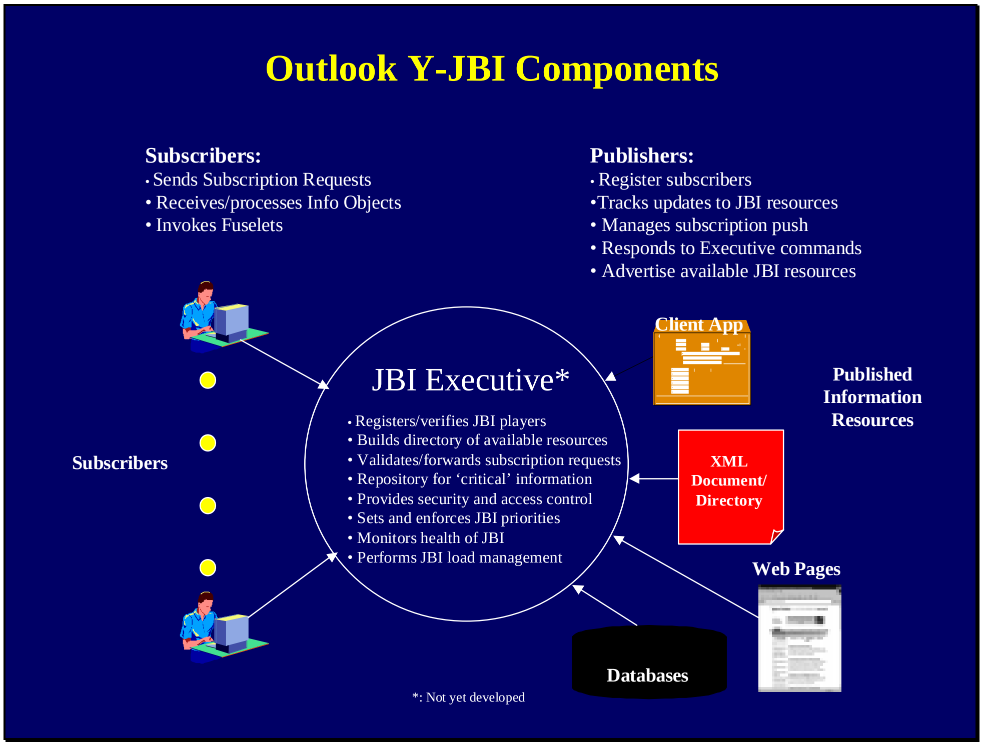


Figure 1 – Performance Statistics for Enterprise E-mail Server [3]

A high-level model of the Outlook Y-JBI architecture is depicted in Figure 2. In the current spiral, P&S operations are accomplished though the peer-to-peer interaction of publisher and subscriber agents. These agents pass XML messages in an E-mail attachment to establish, acknowledge, and forward subscriptions. In this implementation, Microsoft Exchange provides the underlying infrastructure with Microsoft Outlook as the primary e-mail application servicing the user. Outlook was chosen because of its vast user base and its compatibility with a wide array of Microsoft tools and applications. For example, the Collaborative Data Objects (CDO) API enables the construction of fuselets, which seamlessly push the content of Outlook messages to other Microsoft Office applications [**2**]. This effectively creates a direct path for information flow from the subscriber to widely used planning tools such as Excel® and Powerpoint®.

Publisher agents reside on platforms that provide information resources to other Y-JBI users. These agents accept subscription requests from other users, acknowledge them, and send Y-JBI objects to subscribers when the appropriate information resources are updated. Subscription agents process the incoming information objects and trigger fuselets (that reside on the subscriber's system) that process them. The following sections provide an in-depth description of each agent's functionality.



Outlook Y-JBI Components

Subscribers:

- Sends Subscription Requests
- Receives/processes Info Objects
- Invokes Fuselets

Publishers:

- Register subscribers
- Tracks updates to JBI resources
- Manages subscription push
- Responds to Executive commands
- Advertise available JBI resources

JBI Executive*

- Registers/verifies JBI players
- Builds directory of available resources
- Validates/forwards subscription requests
- Repository for 'critical' information
- Provides security and access control
- Sets and enforces JBI priorities
- Monitors health of JBI
- Performs JBI load management

*: Not yet developed

Figure 2 – Outlook Y-JBI Components (Present & Future)

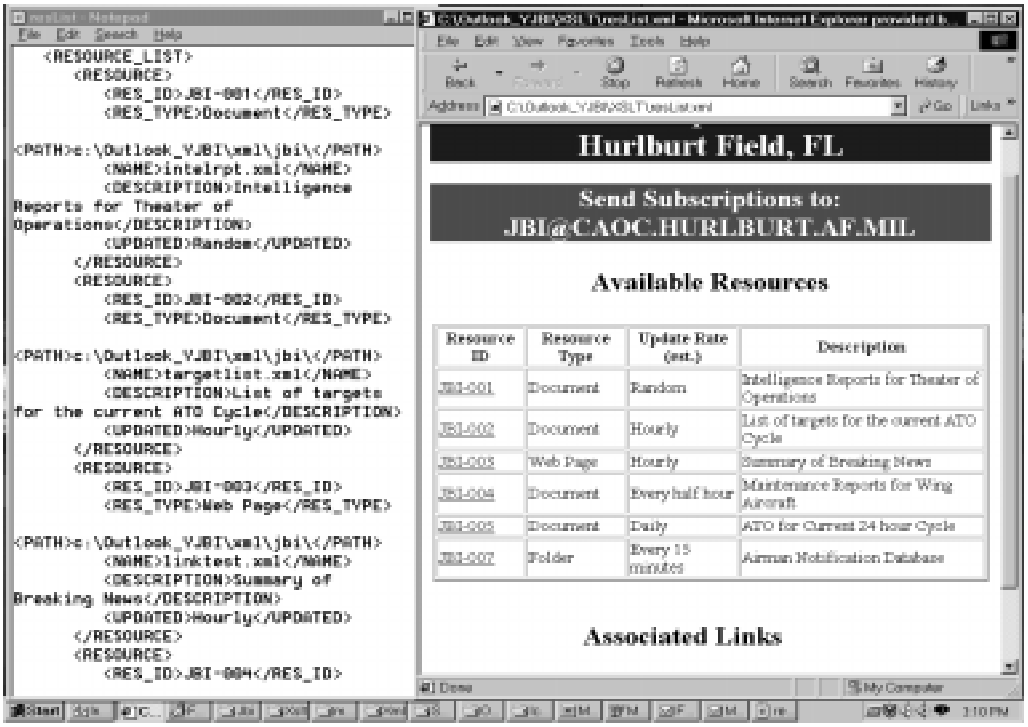
4.1 Publisher Side Processing. Publisher side publishing refers to the functions accomplished by the publisher agent in order to push information objects to subscribers.

Advertising Information Resources. When a publisher connects to the Outlook Y-JBI, it provides a hyperlink to a web page that contains templates of information available to subscribers. This web page can be automatically generated by applying an XSL stylesheet to the XML document containing the list of available resources; Figure 3 shows a sample web page generated in this manner. Users can search these web pages to determine if they want to subscribe to a given resource and download the appropriate template as the basis for creating or modifying subscriptions. Stylesheets for different user categories can be maintained to hide certain resources from unauthorized users.

Accepting Subscription Requests. Upon initialization, the publisher agent validates its resource list and logs into Outlook. If initialization is successful, it can then begin accepting subscription requests from Y-JBI users. The process for handling incoming subscription requests is shown in Figure 4. When a subscription request arrives, the publisher will:

- Validate the format/content of the subscription request.
- Ensure the subscriber is authorized access to the desired resource.

- Register the subscriber by adding the specification contained in the request to the subscription database.
- Reply to the user with the status of the subscription request (accepted/denied/failed).



resList - Notepad
File Edit Search Help

```
   <RESOURCE_LIST>
      <RESOURCE>
         <RES_ID>JBI-001</RES_ID>
         <RES_TYPE>Document</RES_TYPE>

<PATH>c:\Outlook_YJBI\xml\jbi\</PATH>
         <NAME>intelrpt.xml</NAME>
         <DESCRIPTION>Intelligence
Reports for Theater of
Operations</DESCRIPTION>
         <UPDATED>Random</UPDATED>
      </RESOURCE>
      <RESOURCE>
         <RES_ID>JBI-002</RES_ID>
         <RES_TYPE>Document</RES_TYPE>

<PATH>c:\Outlook_YJBI\xml\jbi\</PATH>
         <NAME>targetlist.xml</NAME>
         <DESCRIPTION>List of targets
for the current ATO Cycle</DESCRIPTION>
         <UPDATED>Hourly</UPDATED>
      </RESOURCE>
      <RESOURCE>
         <RES_ID>JBI-003</RES_ID>
         <RES_TYPE>Web Page</RES_TYPE>

<PATH>c:\Outlook_YJBI\xml\jbi\</PATH>
         <NAME>linktest.xml</NAME>
         <DESCRIPTION>Summary of
Breaking News</DESCRIPTION>
         <UPDATED>Hourly</UPDATED>
      </RESOURCE>
      <RESOURCE>
         <RES_ID>JBI-004</RES_ID>
```

C:\Outlook_YJBI\XSLT\resList.xml - Microsoft Internet Explorer provided b...
File Edit View Favorites Tools Help
Back
Forward
Stop
Refresh
Home
Search
Favorites
History
Address
C:\Outlook_YJBI\XSLT\resList.xml
Go
Links
Done
My Computer
Start
3:10 PM

Hurlburt Field, FL

Send Subscriptions to:
JBI@CAOC.HURLBURT.AF.MIL

Available Resources

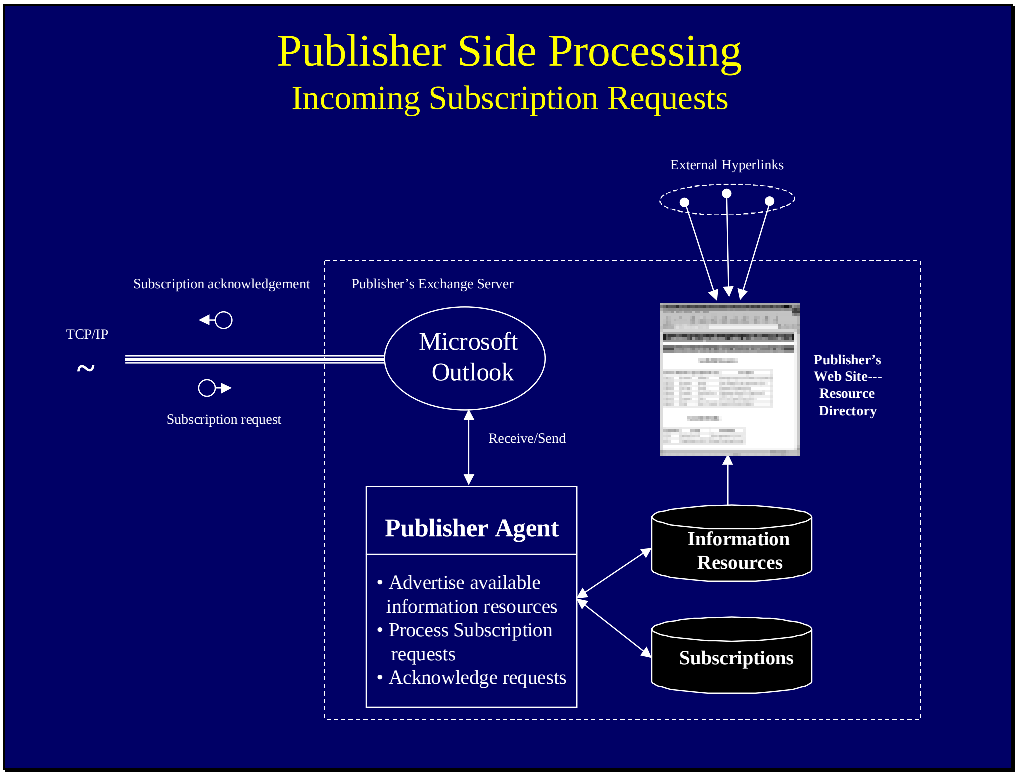
Resource ID	Resource Type	Update Rate (est.)	Description
JBI-001	Document	Random	Intelligence Reports for Theater of Operations
JBI-002	Document	Hourly	List of targets for the current ATO Cycle
JBI-003	Web Page	Hourly	Summary of Breaking News
JBI-004	Document	Every half hour	Maintenance Reports for Wing Aircraft
JBI-005	Document	Daily	ATO for Current 24 hour Cycle
JBI-007	Folder	Every 15 minutes	Airman Notification Database

Associated Links

Figure 3 – XML Resource List Converted to Web Page using XSL Stylesheet

Sending Information Objects to Subscribers. The publisher agent monitors the status of all information resources. When a given resource is updated, the publisher identifies (from the subscriber database) all subscription specifications associated with that resource. This specification is the basis for generating information objects to be sent back to the subscribers. The subscription generation process is shown in Figure 5. For each specification, the publisher will:

- Determine if the *conditions* for receiving the object in the specification are met (start/stop times, refresh rate, logical). Testing the logical conditions involves examining the contents of the information resource. The result is a go/no-go decision as to whether or not to push the object to the subscriber.
- Extract the *subset* of the object requested by the subscriber. In the Outlook Y-JBI implementation, this subset is generated by applying the X-Path query in the subscription request to the XML information resource.
- Tailor the *format* of the information object through the use of XSL scripts. This capability is useful for supporting coalition partners (by filtering out sensitive data) or lightweight devices such as PDAs, cellular phones, or pagers.



Publisher Side Processing

Incoming Subscription Requests

External Hyperlinks

Subscription acknowledgement

Publisher's Exchange Server

TCP/IP

~

Microsoft Outlook

Subscription request

Receive/Send

Publisher's Web Site--- Resource Directory

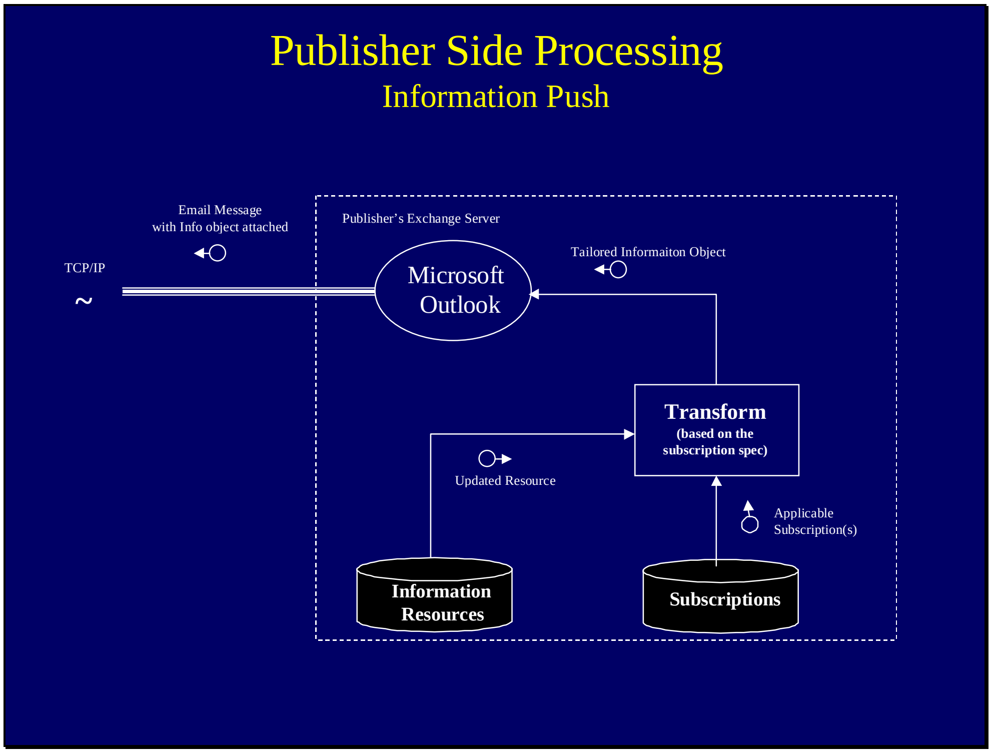
Publisher Agent

- Advertise available information resources
- Process Subscription requests
- Acknowledge requests

Information Resources

Subscriptions

Figure 4 – Subscription Request Processing (Publisher Side)



Publisher Side Processing

Information Push

Email Message with Info object attached

Publisher's Exchange Server

TCP/IP

~

Microsoft Outlook

Tailored Informaiton Object

Transform

(based on the subscription spec)

Updated Resource

Applicable Subscription(s)

Information Resources

Subscriptions

Figure 5 – Information Push (Publisher Side)

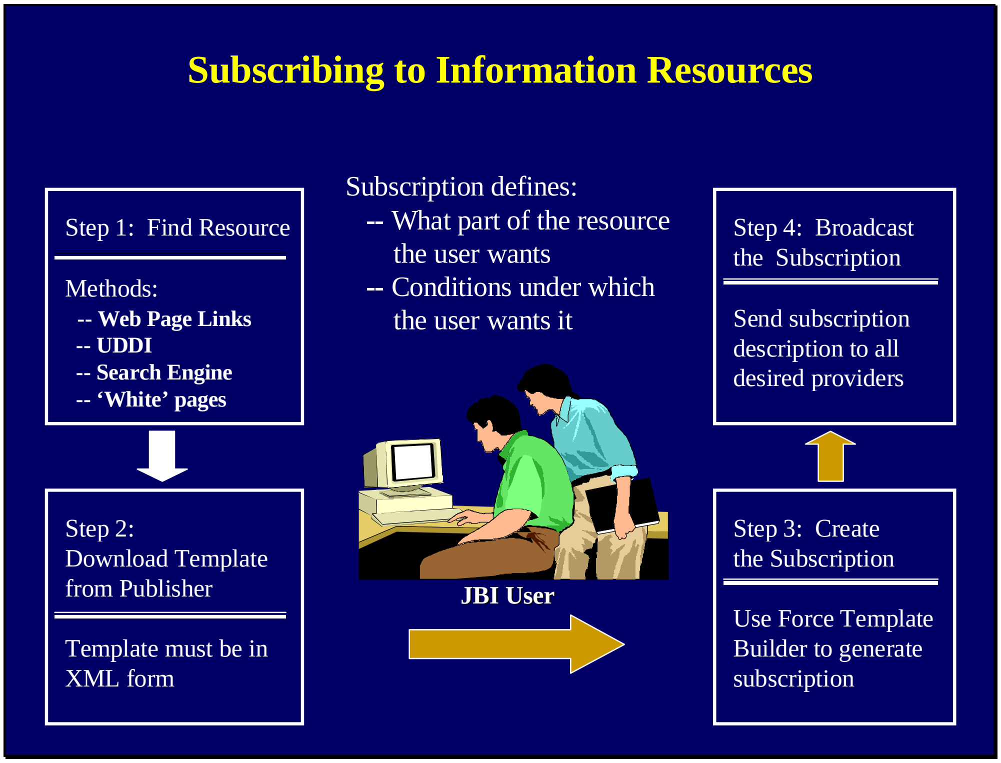
- Generate an XML wrapper (envelope) for the object.
- E-mail the resulting information object back to the subscriber. The object is included in the message as an attachment.

The purpose of much of the processing described above is to minimize the number and size of information objects pushed back to the subscriber. This, in turn, reduces the amount of required communications bandwidth.

4.2 ***Subscriber Side Processing.*** Subscriber side processing refers to function accomplished by the subscriber agent to obtain and process information published by external sources.

Before the subscriber agent can be executed, the user has to first construct a force template. In the Outlook Y-JBI, the force template is merely an XML file consisting of a set of subscription specifications, the associated fuselets, and a mapping between the two. The process for building and activating a force template is described in Figure 6.

Searching for Resources. The first step in building the force template is to search for information resource of interest. Under our scheme, the user locates information resources of interest through a variety of methods. These might include discovery services based on the emerging Universal Description, Discovery, and Integration (UDDI) standard, or simply surfing hierarchical web page links. An important point is that we rely on a man-in-the-loop to ultimately decide what to subscribe to (as opposed to letting an automated information broker make these decisions).



Subscribing to Information Resources

Step 1: Find Resource

Methods:
-- **Web Page Links**
-- **UDDI**
-- **Search Engine**
-- **'White' pages**

Step 2:
Download Template from Publisher

Template must be in XML form

Subscription defines:
-- What part of the resource the user wants
-- Conditions under which the user wants it

JBI User

Step 3: Create the Subscription

Use Force Template Builder to generate subscription

Step 4: Broadcast the Subscription

Send subscription description to all desired providers

Figure 6 – Notional Subscription Process

Creating Subscriptions.

Once a user selects a JBI resource, the next step is to download a template containing the information about the resource needed to create a subscription specification. For a document, directory, or database, this involves giving the user insight into the structure of the resource, and the format and definitions of its data fields. For an application, this involves giving the user insight into the interfaces (input, output) and the processing it performs. The subscription specification is itself an XML document which includes the following information:

- The identity, address, and description of the subscriber
- A unique code for identifying the resource (within the JBI).
- A list of publisher e-mail addresses who the request should be sent to.
- Conditions (logical and temporal) under which the subscription should be send.
- A query (nominally X-Path) for determining what, if any, subset of the information resource should be forwarded to the subscriber.

The key point here is that each subscription specification is tailored for a specific information resource. As part of the Outlook Y-JBI, we have written a Force Template Builder tool that enables a user to construct individual subscription specifications, bind them to fuselets, and combine them into a force template file. Note that the user can elect to construct a force template prior by reusing existing (valid) subscription specifications for a given information resource; this eliminates the overhead of creating force templates from scratch.

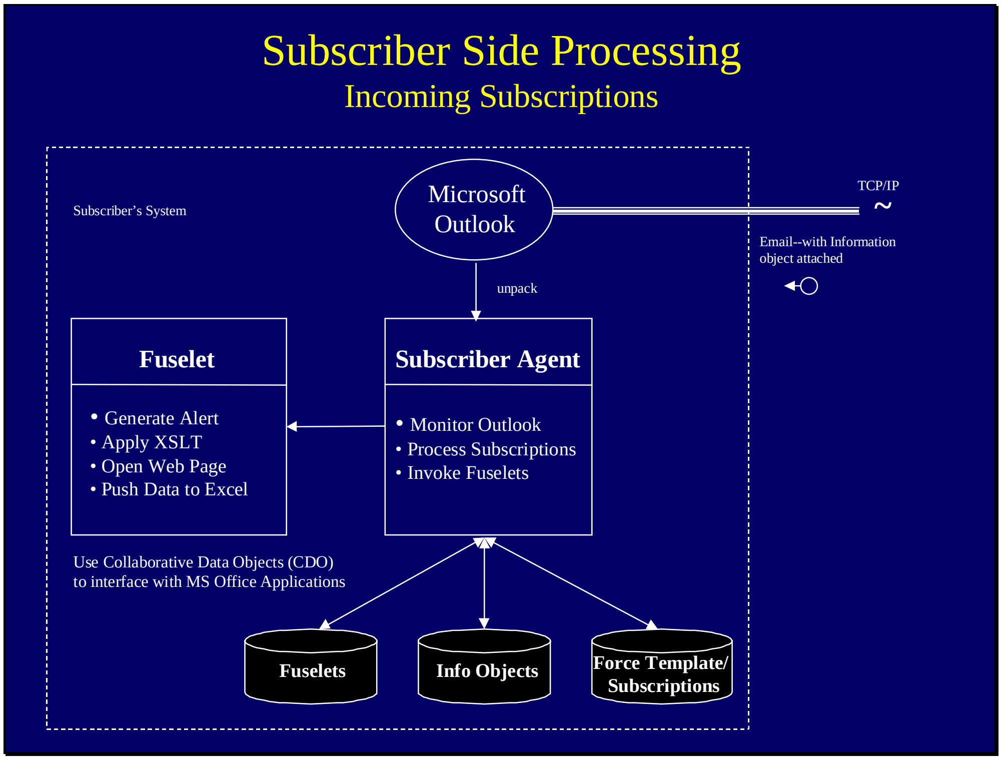
Activating Subscriptions. Upon elaboration, the subscriber agent validates the force template and logs into Outlook. The user can then activate any subset of the subscriptions contained in the force template. The activation generates e-mail to the intended publisher, with the subscription specification included as an attachment. When the acknowledgement is received from the publisher, the subscription status display is updated; this gives the user crucial insight into the status of their requested resources. During the Y-JBI session, the user can elect to suspend, resume, or expunge individual subscriptions. Additionally, the user can selectively enable/disable fuselets.

Invoking Fuselets. As previously discussed, fuselets are utilized to create new knowledge from one or more incoming information objects. The fuselet itself is a short program that extracts specific data from newly arrived information object(s) and processes it in a manner that adds value for the warfighter. Outlook Y-JBI fuselets may be implemented in a variety of ways including:

- As a Visual Basic (VB) program which generates an alert to the user.
- Notification of an updated to a web page of interest (which can then be viewed through a web browser).
- As an XSL script, which displays information on a web browser.
- As a VB program which updates a planner’s Excel spreadsheet.

The power of fuselets is that they can process information that would be tedious and time consuming for the user to do on his own and alert him if anything of importance is discovered. For example, an air mission planner's fuselet might subscribe to ATO and several types of intelligence report. If any new intelligence reports arrive that cover any targets in the ATO (such as a target significantly changing position), then that report can immediately be brought to the attention of the planner. In other cases, fuselets may seamlessly update spreadsheets used by the planner without him being aware of it. Fuselets can also publish new resources to which other users can subscribe. The bottom line is that the Outlook Y-JBI fuselets act solely on the user's behalf and do exactly what the user wants them to.

In the force template defined by the user, the subscriptions are mapped to the specific fuselets that use them. When the information objects corresponding to a given subscription arrive, they are logged against each fuselet. When all the objects needed by a fuselet have arrived, a new fuselet process is spawned. A flowchart describing this data flow is shown in Figure 7.



Subscriber Side Processing

Incoming Subscriptions

Subscriber's System

Microsoft Outlook

TCP/IP

Email--with Information object attached

unpack

Fuselet

- Generate Alert
- Apply XSLT
- Open Web Page
- Push Data to Excel

Subscriber Agent

- Monitor Outlook
- Process Subscriptions
- Invoke Fuselets

Use Collaborative Data Objects (CDO) to interface with MS Office Applications

Fuselets

Info Objects

Force Template/ Subscriptions

Figure 7 – Notional Subscription Process

6. Future Work - Executive Side Processing

Although not part of the current implementation, an Executive component is central to implementing a JBI architecture that is scalable, robust, and secure. The Executive would reside on the JBI platform, which is responsible for providing the core services for the entire infosphere. Note that the JBI platform should not be thought of as one centralized platform, but rather a confederation of distributed systems. When implemented, the Executive will perform a number of necessary, global system management functions and services, including:

Providing security and access controls. User authentication and access control enforcement (authorization) will be accomplished at the Executive level. These controls are essential to the functions described below. Currently, subscription requests are sent directly from a subscriber to the desired publisher(s). Eventually, all subscription requests will be initially sent to the JBI platform. When the Executive approves a request, it will then be forwarded to the appropriate publisher(s).

Registration of JBI components. When JBI components (publishers & subscribers) connect to the system, they will do so through the Executive. In order to register, each component will be required to transmit their force templates to the user. The Executive will be responsible for validating the information contained in the force template and giving the component permission to either publish or subscribe.

Maintaining global directories of available resources. Since the Executive will be privy to all information resources available from connected publishers; it will be able to create the directories that enable Y-JBI users to find those resources. Search services may also be provided to help users locate resources more efficiently.

Maintaining a robust repository. Within the infosphere, there may be some information resources that are so critical or in such high demand, that they assigned to a single publisher. In these cases, the Executive will have the responsibility of replicating these resources in spare repositories.

Enterprise monitoring. The Executive is also charged with monitoring the health of the all components and communications links that make up the JBI enterprise. In addition, the Executive will be able to direct individual components to take corrective action. For example, if a given publisher is generating excessive network traffic, the Executive could direct it to suspend subscribers below a certain threshold priority level.

7. Conclusion

In this paper we have described the current and planned functionality of the Outlook Y-JBI. As stated earlier, we do not expect this e-mail based implementation to evolve into the final JBI architecture. In all likelihood the JBI that is ultimately fielded will support a multitude of transport mechanisms and protocols. Instead, this Y-JBI is a pathfinder that helps AFRL explore the JBI design space. It also demonstrates how existing COTS and W3C compliant freeware can be used to implement JBI-like functionality. Thus, in conclusion, we recap the more noteworthy characteristics of this implementation.

The Outlook Y-JBI is distributed and flexible. The current prototype supports the peer-to-peer interaction of publishers and subscribers. When publishers connect to the Y-JBI, they make a set of information resources available to potential subscribers. For the most part, these resources will reside solely on the publisher's system. This 'repository' is not centralized—rather it is distributed among all attached publishers. As a future safety mechanism, the Executive will subscribe to information of critical importance and, in turn, make these resources available in its own repository. The system is also flexible in that it accommodates the constant arrival/departure of publishers and subscribers.

Centralized control, decentralized execution. When this Y-JBI is fully implemented, the Executive component will provide centralized control of the system for key functions such as security, preservation of critical resources, and load management. However, it will be up to other components to carry out 'directives' issued by the Executive.

Conserving bandwidth is a primary goal. The Outlook Y-JBI maintains a tight coupling between publishers and subscribers. Subscriber agents are kept apprised of the status of their subscriptions and can turn then on or off at will. In turn, publisher agents track specific subscription specifications for a given user. The publisher uses these specifications to tailor the information stream to the exact needs of the subscriber (as opposed to blindly generating unnecessary traffic. By following these guidelines, the architecture minimizes utilization of communication bandwidth.

Fuselets operate on behalf of a given user. A fuselet is a computational entity that creates new information from one or more existing information object. This can include anything from a simple XSL script that transforms an XML document into an HTML document to a client application that builds a theater-wide battlespace picture from multiple data streams. Because they serve a given user (owner), fuselets are only allowed to execute on systems under that user's control. The output of fuselets can, in turn, be published (at the discretion of the fuselet's owner) for the benefit of other subscribers.

References

[1] Jerry Cochran, Tips for Interpreting Messaging Benchmarks, Exchange Administrator (http://www.exchangeadmin.com), April 1999.
[2] Thomas Rizzo, Introduction to Collaborative Data Objects for Microsoft Exchange 2000, Microsoft Developers Network (http://msdn.microsoft.com), Nov 2000.
[3] Microsoft Corporation. Exchange 2000 vs. Exchange 5.5 Scalability Testing, Microsoft Developers Network (http://www.microsoft.com/exchange/techinfo/E2KvsE55.htm), Jan 2001.
[4] Office of the Joint Chiefs of Staff, Joint Vision 2020 (http://www.dtic.mil/jv2020/jvpub2.htm) , 2 Jun 2000.
[5] United States Air Force Scientific Advisory Board, Report on Information Management to Support the Warrior, SAB-TR-98-02 (http://www.sab.hq.af.mil/Archives/index.htm), December 1998.
[6] United States Air Force Scientific Advisory Board, Report on Building the Joint Battlespace Infosphere, SAB-TR-99-02 (http://www.sab.hq.af.mil/Archives/index.htm), December 1999.