J12.5 AN EXAMPLE OF NEXTGEN WEATHER INFORMATION INTEGRATION AND MANAGEMENT

Russ Sinclair*, Tom Hicks, Carlos Felix, Keith Bourke Harris Corporation, Melbourne, Florida

1. INTRODUCTION

In the NextGen era weather information will be managed as a net-centric, virtual, data repository, referred to as the "four-dimensional Cube (4D Cube)". The 4D Cube will be "realized" and made accessible to National Air Space (NAS) users via a Service Oriented Architecture (SOA). COTS SOA software will be augmented with custom software to implement the NAS SOA architecture. To anticipate and mitigate some of the issues NextGen weather domain implementers may encounter, Harris created a SOA-based NextGen weather information management model in our Research and Development (R&D) lab. This paper describes our weather information management environment and reports on experiences using Oracle-based JAVA Messaging Service (JMS) publish/subscribe (pub/sub) and Enterprise Service Bus (ESB) capabilities to share weather information among weather information producers and consumers. While our environment includes request/response web services in addition to pub/sub services, our focus in this paper is on pub/sub. The following topics will be covered:

- Key Architectural Concepts – first a few architectural concepts are defined to help readers unfamiliar with SOA concepts better understand the paper
- Project Approach – describes our basic project strategy
- Weather Information Management Lab Environment – description of the legacy and new software applications comprising our lab environment
- Information Flow – description of several of the key information exchange scenarios exercised using pub/sub messaging
- 4D Cube Simulation – description of how we developed and managed our 4D Cube
- Conclusions – description of some of our more interesting lessons-learned
- Future Plans – description of the next steps planned for our R&D program

---

* *Corresponding author address:* Russ Sinclair, Harris Government Communications Systems Division, P.O. Box 9800 Melbourne, FL 32902-9800; e-mail: russell.sinclair@harris.com

2. KEY ARCHITECTURAL CONCEPTS

This section defines a few commonly used architectural concepts that are referenced and discussed throughout this paper.

Service Oriented Architecture

- An architectural approach that eliminates single-purpose applications and point-to-point communications between integrated applications and replaces them with loosely coupled reusable business and IT applications, called services
- COTS and Open Source software is available to facilitate the implementation of SOA (e.g., IBM, Oracle, FUSE, etc.)
- Services can support flexible information processing and exchange among data producers and consumers
- Registries are used to publish and discover services and service artifacts
    - o The Universal Description Discovery and Integration (UDDI) and the Web Services Description Language (WSDL) are Web Service Registry Standards
    - o Electronic Business using eXtensible Markup Language Registry (ebXML) is a Registry Standard
- An ESB is a key SOA component that provides fundamental services for complex architectures via an event-driven and standards-based messaging-engine

Pub/Sub Messaging

- A messaging pattern where message **producers** (publishers) send messages to a broker that forwards messages to **consumers** (subscribers) based on each consumer's registered interest (subscription)
- Publishers and subscribers require no knowledge of each other, as this responsibility is handled by the broker
- Topics are "groups" of messages to which consumers subscribe
- Consumers can subscribe to one or more topics
- All subscribers to a topic will receive the same messages

- Often used to deliver information that continually changes, like the “news”, in a package that remains constant, like the “newspaper” – applicable to many kinds of weather information products, e.g., numerical model forecasts

Request/response

- A messaging pattern where producers offer on-demand services for one time delivery of information to consumers
- Web services in conjunction with a run-time Service Registry (e.g., UDDI) are utilized to provide a standards based approach

Note that both pub/sub and request/response patterns are needed since not all information exchange can be effectively accomplished using only one or the other pattern.

3. PROJECT APPROACH

Harris prototypes developed in our Weather and Network R&D programs were used to create a NextGen-like SOA environment for the evaluation of 4D Cube producer and consumer information exchange. The Data Exchange (DEX), a Harris prototype for the FAA’s future System Wide Information Management (SWIM) core services, provided the SOA backbone. Both legacy and new weather applications were used to simulate the 4D Cube and the DEX was used to “connect” consumers to 4D Cube products. Details of the applications used in the lab are discussed in the next section.

4. WEATHER INFORMATION MANAGEMENT LAB ENVIRONMENT

A block diagram of the Weather Information Management Model in our lab is shown in Figure 1. The seven blocks in Figure 1 represent Harris Weather and Network R&D components leveraged to model the information producers and consumers and the SOA technology that will loosely couple consumers and producers in the NextGen era.

Arrows from the consumer and producer blocks represent the publication and subscription information exchanged between the consumers and producers and the DEX needed to set up the run-time pub/sub information exchange.
The seven components are :

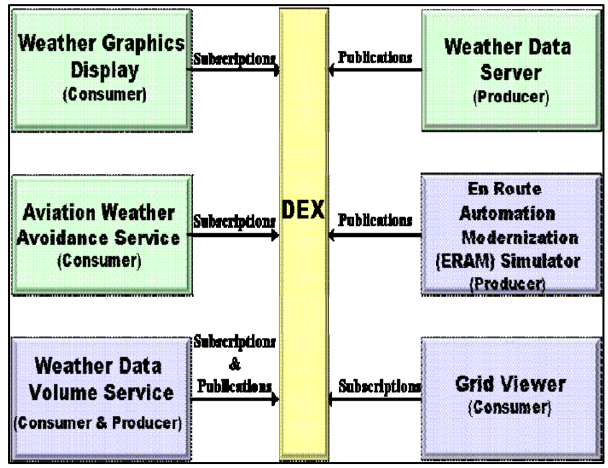

Figure 1. Weather Information Management Model

Data Exchange (DEX)

Shown at the center of Figure 1, the DEX is comprised of Oracle and custom software that provides SWIM Core (SOA) services including messaging, security, enterprise management, and interface management. In our model, the DEX facilitates data exchange among simulated 4D Cube producers and consumers. It is the heart of our 4D Cube and contains the message broker used to de-couple producers and consumers.

Weather Data Server (Producer)

Shown in the upper right hand corner of Figure 1, the Weather Data Server is a consolidated view of several Harris legacy applications that ingest weather products from the NWS, the GOES satellites and the Internet, and continually stream them to other applications. In our model, the Weather Data Server represents NWS and other key weather information producers that will provide 4D Cube weather products to NAS consumers. The legacy Weather Data Server applications were modified to publish “4D Cube” products to the DEX.

Weather Graphics Display (Consumer)

Shown in the upper left corner of Figure 1, the Weather Graphics Display is a legacy display application that ingests weather products (e.g., NWS and GOES satellite products) & transforms them into graphical displays. In our model the Weather Graphics Display simulates a NextGen software application consumer that requires 4D Cube inputs. The legacy Weather Graphics Display application was modified to subscribe to 4D Cube products offered by the DEX. Although NextGen 4D Cube consumers are not expected to display weather graphics in the long-term, this legacy application was used in our model because it visually demonstrates the information exchange facilitated by the DEX.

En Route Automation Modernization (ERAM) Simulator (Producer)

Shown below the Weather Data Server in Figure 1, the En Route Automation Modernization (ERAM) Simulator generates simulated 4D flight trajectories and provides them to other applications by publishing them to the DEX. In our model, it simulates an ERAM-like data source application in the NextGen timeframe.

Aviation Weather Avoidance Service (Consumer)

Shown below the Weather Graphics Display in Figure 1, the Aviation Weather Avoidance Service ingests select 4D Cube weather products, e.g., convective weather forecasts, as well as simulated 4D flight trajectories. It assesses the 4D weather impact on the simulated 4D flight trajectories and suggests alternate flight routes when pre-defined weather impact thresholds for a trajectory are exceeded. This application can be configured to process up to 5000 flights every 5 min. In our model, the Aviation Weather Avoidance Service was used to represent several different types of NextGen consumers: an Air Navigation Service Provider (ANSP) consumer, which would monitor all CONUS flight trajectories for impacts; an Airline Operations Center (AOC) consumer, which is interested in impacts to all the airline's flight trajectories; and, a single pilot consumer, interested in impacts to their flight trajectory.

Weather Data Volume Service (Consumer & Producer)

Shown below the Aviation Weather Avoidance Services in Figure 1, the Weather Data Volume Service ingests RUC grid analyses/forecasts and subsets (crops) them in accordance with predefined 2D geographical windows. When created, the cropped RUC grids are published to the DEX. In our model, this application represents a NextGen data service application that tailors 4D Cube gridded data products for consumers with specific geographic coverage requirements that remain consistent over time.

The Weather Data Volume Service also has a Web Services request/response interface that allows consumer applications to request cropped RUC grids for ad hoc geographical windows.

Grid Viewer (Consumer)

Shown below the ERAM Simulator in Figure 1, the Grid Viewer subscribes to cropped RUC grids published to the DEX. In our model the Grid Viewer simulates a NextGen software application consumer that has specific RUC Grid geographic coverage requirements that remain constant over time. Although NextGen 4D Cube consumers are not expected to display weather graphics in the long-term, this grid display application was used to demonstrate the cropped RUC grid exchange facilitated by the DEX.

The Grid Viewer also has a web client interface that enables a user to make requests for RUC grids cropped to ad hoc geographical windows.

5. INFORMATION FLOW

Key examples of information flow used to evaluate pub/sub information exchange are depicted by the numbered, color coded arrows in Figure 2. Each color/number depicts a different pub/sub flow.

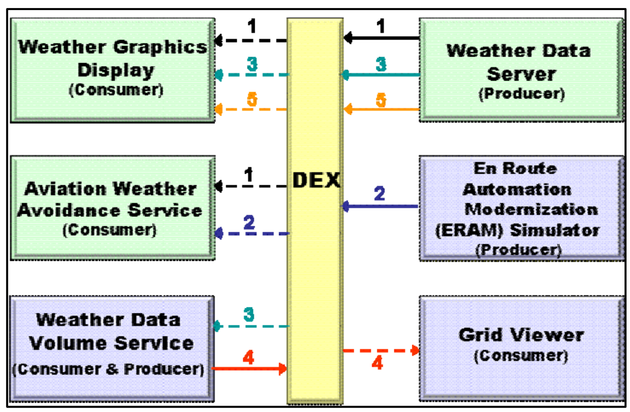

Figure 2. Key Information Flows

The solid arrows annotated with numbers depict the delivery of information from the consumer to the DEX, i.e., the publication of information. The dashed arrows annotated with numbers depict the delivery of information from the DEX to the consumers, i.e., the fulfillment of consumer subscriptions.

The five flows highlighted in Figure 2 are:

- Flow 1 - represents the pub/sub delivery of National Convective Weather Forecast (NCWFs) products from the Weather Data Server to both the Weather Display and Aviation Weather Avoidance Server applications. NCWF products, produced every five minutes by the NWS Aviation Weather Center, were ingested by the

Weather Data Server and published to the DEX. The DEX, fulfilling subscriptions for the NCWF, pushed them to the Weather Graphics Display and the Aviation Weather Avoidance Service.

- Flow 2 - represents the pub/sub delivery of simulated ERAM flight trajectory objects to Aviation Weather Avoidance Service. A batch of simulated ERAM flight trajectories were generated every five minutes and published to the DEX. The DEX, fulfilling subscriptions for the flight trajectories, pushed them to the Aviation Weather Avoidance Service.
- Flow 3 - represents the pub/sub delivery of RUC grids to both the Weather Display and Weather Data Volume Service. RUC grids, produced every hour by the National Center for Environmental Prediction, were ingested by the Weather Data Server and published to the DEX. The DEX, fulfilling subscriptions for the RUC grids, pushed them to the Weather Graphics Display and the Weather Data Volume Service.
- Flow 4 - represents the pub/sub delivery of cropped RUC grids to the Grid View application. RUC cropped grids were generated for predefined 2D geographical windows over CONUS, upon the receipt of their "parent" RUC grids (Flow 3). Each time a cropped RUC grid was generated, it was published to the DEX. The DEX, fulfilling subscriptions for the cropped RUC grids, pushed them to the Grid Viewer application.
- Flow 5 – represents the pub/sub delivery of a large number of products from the Weather Data Server to the Weather Display application. This flow is analogous to the point to point weather product streaming traditionally seen between weather data servers and weather display systems that generate and display a multitude of different types of weather graphics.

6. 4D CUBE SIMULATION

A key objective of our approach was to simulate the NextGen 4D Cube and provide a mechanism for the consumer/user to visualize the 4D weather information available to them and select products of interest. In our model the NextGen 4D Cube was simulated by all the weather products published to the DEX from the Weather Data Server and the Weather Data Volume Service. To aid the consumer/user in visualizing the 4D Cube, a Product Catalog concept was formulated. Our Product Catalog concept is analogous to a library card catalog that contains the names of (published) books and short compilations of metadata about the books, e.g., author names, publication dates, etc. The Product Catalog can be browsed and searched to aid a user in discovering products of interest. In our Product Catalog implementation, only a small amount of metadata was used to enable simple searches. In the NextGen era, such a product catalog would need to contain a rich set of metadata and sophisticated search tools to enable the effective discovery and exploitation of the 4D Cube products.

Although a UDDI Registry is specifically designed for advertising web services, not published products, we attempted to leverage our COTS Oracle UDDI Registry to implement the Product Catalog. After a couple false starts we abandoned use of the UDDI Registry in favor of a custom approach based on an Oracle relational data base. We recognize that an ebXML-based registry would provide an effective basis for such a Product Catalog. However, our Oracle based suite of tools did not offer an ebXML registry, and for expedience, we chose the custom approach.

A display tool, called the DEX Navigator, was developed to facilitate 4D Weather Cube Product Catalog browsing by consumers and subscription management by the DEX administrator. The DEX Navigator enables weather product consumer users to visualize the 4D Cube as a list of products that can be selected for automatic delivery using the pub/sub messaging pattern. It presents the viewer with a list of products and their associated metadata and allows them to build a subscription containing multiple products using drag and drop actions. Figure 3 highlights how the 4D Cube is created, managed and visualized.

The DEX Administrator physically creates/updates the Product Catalog which identifies all the products published "to the Cube". As shown in the upper half of Figure 3, the DEX Administrator receives product names and metadata of 4D Cube contributors from the producer developer. The DEX Administrator manually adds the product names and metadata to the DEX Weather Product Catalog, "identifying" the components of the 4D Cube to the DEX. In an operational environment, the DEX Administrator would likely require authorization from the Weather Domain Authority, and maybe even authorization from the Enterprise Domain Authority, to load a producer's Product IDs and associated metadata into the Weather Product Catalog.

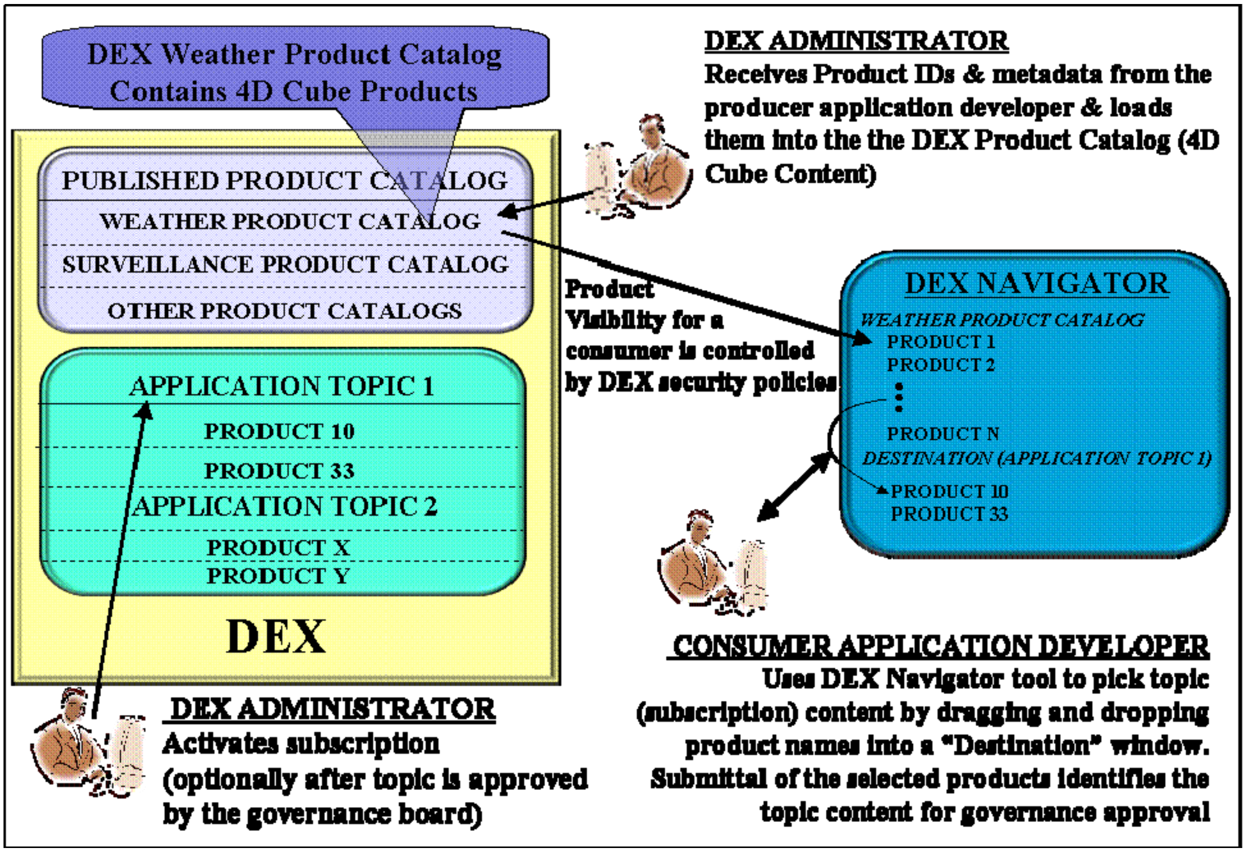

Figure 3. Interactions with the Weather Product Catalog and Subscription

While the producer developer and DEX administrator actions to publish products are currently manually intensive, they can and should be automated in the real world environment by tools like the DEX Navigator. As shown in Figure 3, the Weather Product Catalog can be one of many Product Catalogs built and managed by the DEX Administrator. Minimally, a Product Catalog for each NAS application domain (e.g., Surveillance) is envisioned.

The DEX Navigator displays the Weather Catalog to the consumer user in a window. The Weather Catalog product visibility and access is managed by the DEX in accordance with the consumer's access authorization policies; only those products authorized to be accessed by a particular user are displayed in the DEX Navigator Product Catalog Window.

Illustrated on the right side of Figure 3, the consumer application developer constructs a subscription by dragging and dropping product names from the DEX Weather Product Catalog Window into a Window titled the DESTINATION. The name assigned to the DESTINATION is pre-established by the DEX administrator for each consumer. The "DESTINATION" (name) is like a Topic in pub/sub parlance.

As illustrated in the lower left portion of Figure 3, subscription product selections made by the consumer application developer using the DEX Navigator are activated in the DEX by DEX Administrator action. Optionally this could require approval by the governance board.

7. CONCLUSIONS

Key conclusions from our work modeling the 4D Cube from a pub/sub perspective are:

- Due to the very large number of weather products contributing to the 4D Cube, careful consideration must be given to how published products are grouped as topics for subscription

- Topic design directly affects infrastructure resource utilization by COTS SOA
- Constraining the number of topics to an optimal number will likely be necessary and this requires a good understanding of the active number of consumers and their product needs
  - Too many topics increased development and maintenance complexity and adversely affected software initialization performance
    - While we did not measure operational performance, our inference is that added overhead would require additional runtime resources and when capacity becomes strained would adversely affect runtime performance
  - An a priori mapping of products to a limited number of topics streamlined development and maintenance and resulted in good software initialization performance, but caused consumers to receive unwanted products and wasted LAN bandwidth
  - Allocation of one topic per consumer with consumer choice of products at fine granularity, streamlined development and maintenance, resulted in good software initialization performance, and optimized user product selection and LAN bandwidth utilization, however,
    - If a large number of consumers are expected, the same issues surrounding the use of too many topics would prevail
- COTS products will likely require customization and augmentation with additional custom developed software to facilitate effective 4D Cube creation/management and flexible consumer access
  - Our COTS JMS pub/sub and ESB capabilities did not offer a straightforward way to visualize the 4D Cube, i.e., discover and access 4D Cube products
  - A custom product catalog based on a relational DB was developed
- A pub/sub jumpstart toolkit developed to enable rapid on-ramping (interfacing to the DEX) of producers and consumers proved to be very useful

8. FUTURE PLANS

Next steps with our SOA-based NextGen weather information management model include incorporation and evaluation of:

- OGC standards
- Web services request/response for 4D Cube products
- Advanced radar mosaic processing