



Aviation Mashups

May 13, 2009

Samet Ayhan, Paul Comitz, The Boeing Company

Vladimir Stemkovski, IBM

Contents

Boeing Air Traffic Management

- **Problem Statement**
- **Objective**
- **Overview**
- **Scenario**
- **Creating Data Feeds**
- **Creating Widgets**
- **Assembling the Mashup Application**
- **Conclusions**

Problem Statement

Boeing Air Traffic Management

- One of the challenges facing the Aviation community is not having a one-stop-shop environment where user can perform searches by connecting to multiple disparate data sources
- What if a user wants to look for airports with certain attributes in a particular region along with all crucial aviation services?
- What if he wants to see all results on a single web page that displays not only attribute data on a tabular form but also on a map with weather data overlaid?

Problem Statement Cont'd

Boeing Air Traffic Management

- A common approach to this problem would be opening up multiple browsers, visiting each services separately, making requests per service, gathering outputs and trying to make sense out of results. In fact, there still won't be an output page that fuses the results together on a single page.
- To address this practice, Boeing Advanced Air Traffic Management (AATM) and IBM JStart Team implemented a Web 2.0 Mashup application

Objective

Boeing Air Traffic Management

- The inspiration came from interviews with subject matter experts and operational practitioners. These interviews were focused on aviation operations in the presence of natural disasters such as the hurricanes along the Gulf Coast in 2005. Government aviation operations managers advised the NEO team of a request made by the Secretary of Transportation during one of the disasters. The secretary asked the operations managers
 - If they could identify any open airports in the hurricane affected areas.
 - If specific characteristics of each airport (runway length) could be identified.This information would be used for search and rescue operations.

Overview

Boeing Air Traffic Management

- Providing interagency collaboration through the use of an underlying networked architecture was the objective of NEO Spiral One program.
- For this effort, information from different agencies such as DoD, DHs, DoT including FAA needed to be brought together.

Scenario

Boeing Air Traffic Management

- A natural disaster has occurred in an area of the US and the government is trying to organize an appropriate relief effort. One aspect of the government's response is to fly in desperately needed supplies on C130 aircrafts. C130's have special airport runway requirements and cannot land at just any airport.
- If the natural disaster was severe and widespread, such as resulting from a category 5 hurricane, many airports in the effected region may be damaged and unable to land planes, even if the runways were long enough.

Scenario Cont'd

Boeing Air Traffic Management

- This mashup application will help answer the question of what airports are open, which ones can a C130 land at, and will provide other pertinent information to the user, someone who is helping to organize and coordinate the relief effort. Today, this information is all over in multiple sources and agencies.
- How is this information brought together on a single web page, providing the user with tools and data to make smart decisions on a timely manner?

IBM Mashup Center

Boeing Air Traffic Management

- IBM Mashup Center addresses this need by providing two main components:
 - InfoSphere Mashup Hub: The platform that contains visual tools for exposing both enterprise and departmental data sources as data feeds as well as allowing transformation and remixing of these data feeds so they could be easily consumed within mashup applications.
 - Lotus Mashups: The graphical, browser-based environment in which users assemble and wire programmable and reusable components called widgets into mashup applications.

Creating Data Feeds

Boeing Air Traffic Management

- Data feeds were created by connecting to various data sources. Database and Web Services are data sources used for this application. Below are the data feeds created:
 - Airport List
 - Airport details
 - Runway details
 - Airport status
 - NOTAM Web Service
 - METAR Web Service
 - RUC Web Service

Airport List Data Feed

Boeing Air Traffic Management

- This feed connects to the database and makes a query by joining two tables:
 - Airportstatus and
 - Runway

Below is the full SQL statement to retrieve data:

```
SELECT arpt.ARPT_IDENT, arpt.NAME, arpt.ICAO, arpt.FAA_HOST_ID,  
arpt.WGS_DLAT, arpt.WGS_DLONG, (select max(LENGTH) from runway  
where runway.ARPT_IDENT = arpt.ARPT_IDENT) as  
MAX_RUNWAY_LENGTH, (select count(1) from runway where  
runway.ARPT_IDENT = arpt.ARPT_IDENT) as RUNWAYS FROM  
icaovsairport arpt WHERE (arpt.WGS_DLAT between ':lat1' AND ':lat2') AND  
(arpt.WGS_DLONG between ':long1' and ':long2') ORDER BY arpt.NAME  
ASC
```

Airport Details Data Feed

Boeing Air Traffic Management

- This feed connects to the database and makes a query on the following table:
 - Icaovsairport

Below is the full SQL statement to retrieve data:

```
select ICAOVSAIRPORT.NAME, ICAOVSAIRPORT.ICAO,  
ICAOVSAIRPORT.FAA_HOST_ID, ICAOVSAIRPORT.WGS_DLAT,  
ICAOVSAIRPORT.WGS_DLONG, ICAOVSAIRPORT.ELEV,  
ICAOVSAIRPORT.TYPE from ICAOVSAIRPORT where  
ICAOVSAIRPORT.ARPT_IDENT = ':arpt_ident' order by  
ICAOVSAIRPORT.ARPT_IDENT asc
```

Runway Details Data Feed

Boeing Air Traffic Management

- This feed connects to the database and makes a query on the following table:
 - Runway

Below is the full SQL statement to retrieve data:

```
select RUNWAY.LENGTH, RUNWAY.WIDTH, RUNWAY.SURFACE from  
RUNWAY where RUNWAY.ARPT_IDENT = ':airport_code' order by  
RUNWAY.HIGH_IDENT asc
```

Airport Status Data Feed

Boeing Air Traffic Management

- This feed connects to the database and makes a query on the following table:
 - Airportstatus

Below is the full SQL statement to retrieve data:

```
select RUNWAY.LENGTH, RUNWAY.WIDTH, RUNWAY.SURFACE from  
RUNWAY where RUNWAY.ARPT_IDENT = ':airport_code' order by  
RUNWAY.HIGH_IDENT asc
```

NOTAM Data Feed

Boeing Air Traffic Management

- This feed connects to the NOTAM Web Service by utilizing its WSDL. Given 4-letter airport identifier, ICAO, the web service returns NOTAM alerts. Below is the method and its signature:

```
public NotamMessage[] getNotamsMessage(java.lang.String ICAO) throws  
java.rmi.RemoteException;
```

METAR Data Feed

Boeing Air Traffic Management

- This feed connects to the METAR Web Service by utilizing its WSDL. Given 4-letter airport identifier, ICAO, the web service returns METAR report. Below is the method and its signature:

```
public String getMetarReport(java.lang.String ICAO) throws  
java.rmi.RemoteException;
```

RUC Data Feed

Boeing Air Traffic Management

- This feed connects to the RUC Web Service by utilizing its WSDL. Given 4-letter airport identifier, ICAO, the application queries the airports database and receives the latitude, longitude values for the airport. Using the location data, the application invokes the web service method with 4-hour forecast:

```
public String getMashupReport(double latitude, double longitude, int  
forecastHour) throws java.rmi.RemoteException;
```

Corridor Integrated Weather Service (CIWS) Image

Boeing Air Traffic Management

- Created at MIT Lincoln Lab, CIWS integrates data from national weather radars with thunderstorm forecasting technology to aid analysis of airspace congested by convective weather.
- CIWS Atom feed client downloads the current CIWS VIL and Echo Top data, converts it from the native Lambert Conformal conic map projection into a Mercator map projection in the form of an image and makes it available for Mashup application.

Creating Widgets

Boeing Air Traffic Management

- Widgets are programmable, reusable simple browser oriented components that provide either a logical service to the page or visualization for the user. Widgets can be also wired together and exchange events to address a specific business case. Widgets do not need to be aware of each other before the mashing (wiring) occurs. While some widgets are visible on the web page, others are invisible, performing certain tasks, with no graphical user interface.

Creating Widgets Cont'd

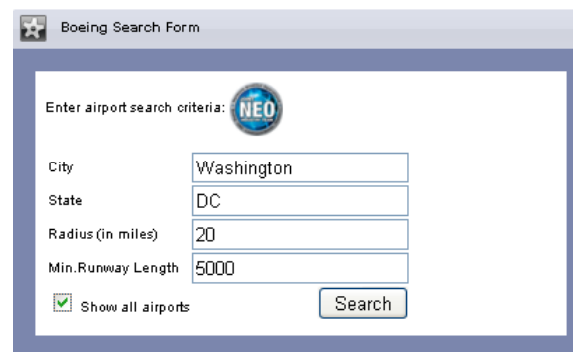
Boeing Air Traffic Management

- The following widgets were implemented to address the business case:
 - Search Form
 - Airports Search
 - Airports Status
 - Runways Search
 - CIWS Image
 - Open Street Map
 - NOTAM Search
 - METAR Search
 - RUC Search

Search Form Widget

Boeing Air Traffic Management

- Allows the user to enter parameters in text fields and fires events. The parameters are City, State, Radius and Minimum Runway Length. City and State fields define the center of the search area. Radius is the distance in miles from the center defining the circular area. Minimum Runway Length (in feet) is the criterion for searching the airports



The screenshot shows a web browser window titled "Boeing Search Form". Inside the window, there is a search form with the following elements:

- A heading: "Enter airport search criteria:" followed by a circular logo with the letters "NEO".
- A "City" text input field containing the text "Washington".
- A "State" text input field containing the text "DC".
- A "Radius (in miles)" text input field containing the text "20".
- A "Min.Runway Length" text input field containing the text "5000".
- A checkbox labeled "Show all airports" which is checked.
- A "Search" button.

Search Form Widget Cont'd

Boeing Air Traffic Management

```
BoeingSearchForm.js - WordPad
File Edit View Insert Format Help

searchAirports: function() {
    var city = dojo.string.trim(dojo.byId(this.id + "cityField").value);
    var state = dojo.string.trim(dojo.byId(this.id + "stateField").value);
    var radius = dojo.string.trim(dojo.byId(this.id + "radiusField").value);
    var minRunwayLength = dojo.string.trim(dojo.byId(this.id + "runwayField").value);
    var isShowAllAirports = dojo.byId(this.id + "showAllCheckbox").checked;

    if (parseFloat(radius) > this.defaultRadiusValue)
    {
        this.radiusValue = radius;
    }
    else
    {
        this.radiusValue = this.defaultRadiusValue;
    }

    var isValid = this.validateForm(city, state, radius, minRunwayLength);
    if (!isValid) {
        return;
    }
    else {
        var location = { "city" : city,
                        "state" : state,
                        "radius" : radius};

        this.iContext.iEvents.fireEvent("location",
                                        "locationInfo",
                                        location);

        this.iContext.iEvents.fireEvent("runwaylength",
                                        "string",
                                        minRunwayLength);

        this.iContext.iEvents.fireEvent("showAllAirports",
                                        "string",
                                        isShowAllAirports);

        var attributesSet = this.iContext.getWidgetAttributes();
        var useLocalWeatherImages = attributesSet.getItemValue("useLocalWeatherImg");

        if (useLocalWeatherImages == "false" || useLocalWeatherImages == false) {
            this.getGeocode(city + " " + state);
        } else {
            this.createOverlay({"longitude" : "-81.309", "latitude": "28.679"});
        }
    }
},
},
```

Airport Search Widget

Boeing Air Traffic Management

- Is hidden and invokes a MashupHub data feed to get a list of airports in the US. The widget then filters the data based on the “radius” value that was entered in the search form. The widget performs the following tasks :
 - Invokes a geocoding service to get the latitude and longitude values of the city and state that was entered.
 - Using the “radius” value that was entered, determines what airports fall inside of this circular area .

Airport Status Widget

Boeing Air Traffic Management

- Is hidden and it listens for the “airports” event and invokes a MashupHub data feed to get the status of each airport in the list. The widget also listens for the “Show all airports” event from the Search Form widget.

Runways Widget

Boeing Air Traffic Management

- Listens for the “open airports” event from the Airport Status widget and the “runway length” event from the Search Form widget. For each airport on the list, the widget invokes the runways feed to determine which airports have long enough runways to meet the user entered “minimum runway length” criterion

CIWS Image Widget

Boeing Air Traffic Management

- Included inside of Search Form widget. As soon as the user clicks “Search” button, the widget gets the current CIWS image by invoking a web service and overlays on top of the Open Street Map. The image is in GIF format and supports transparency. The image is in Mercator projection accurately matching the underlying map’s geographic position. It covers the Continental US.

Open Street Map Widget

Boeing Air Traffic Management

- The OSM widget listens for “open airports with usable runways”, “open airports without usable runways” and “closed airports” events from the AirportsStatus and Runways widgets and plots the location of all the airports published on the map. Airports from the “open airports with usable runways” array have an airport icon surrounded by green, airports from the “open airports without usable runways” array have an airport icon surrounded by yellow while airports from the “closed airports” array have an airport icon surrounded by red. When one of the airport icons is clicked, a “map data” event is fired, publishing the data associated with that particular airport.

NOTAM Search Widget

Boeing Air Traffic Management

- Hidden and invokes a MashupHub feed to get the list of “NOtices To AirMen” for a particular airport. The MashupHub feed is sourced by a SOAP based web service. The widget listens for MapData events and extracts the airport code from the map data and then invokes the feed URL, passing the airport code (ICAO) and any other parameters required by the feed.

METAR Search Widget

Boeing Air Traffic Management

- Hidden and invokes a MashupHub feed to get the “Metar Report” for a particular airport. The MashupHub feed is sourced by a SOAP based web service. The widget listens for MapData events and extracts the airport code from the map data and then invokes the feed URL, passing the airport code (ICAO) and any other parameters required by the feed.

RUC Search Widget

Boeing Air Traffic Management

- Hidden and uses airport latitude and longitude to invoke a MashupHub feed to get the Ruc data. The MashupHub feed is sourced by a SOAP based web service. The widget listens for MapData events and extracts the airport code from the map data and then invokes the feed URL, passing the airport coordinates data required by the feed

Data View Widget

Boeing Air Traffic Management

- Out-of-the-box Lotus Mashups widget.
- Displays data in tabular form.
- Used to display a list of open airports with usable runways as well as various data associated with an icon selected by the user on the map. When the icon is clicked by the user, this widget displays the details of the data associated with the airport that was selected.



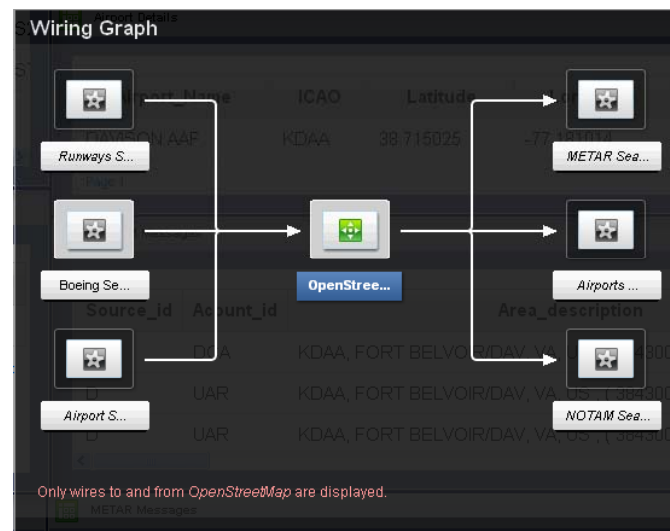
The screenshot shows a window titled "NOTAM Messages" containing a data table. The table has the following columns: Source_id, Acount_id, Area_description, Notam_id, Part, Lastmod, and an unlabeled column. The data rows are as follows:

Source_id	Acount_id	Area_description	Notam_id	Part	Lastmod	
D	UAR	KIAD, WASH DC/DULLES, VA, US, (385559N, 0772700W) [IAD]	02/015	1/1	200802211824	AIRSPACE
D	IAD	KIAD, WASH DC/DULLES, VA, US, (385559N, 0772700W) [IAD]	02/002	1/1	200802011522	TWY E CL
D	IAD	KIAD, WASH DC/DULLES, VA, US, (385559N, 0772700W) [IAD]	08/002	1/1	200908021207	TWY K7 C

Assembling The Mashup Application

Boeing Air Traffic Management

- Once the data feeds are created and widgets are developed, the user is ready to assemble the application in Lotus Mashups environment. Assembling process includes creating a new page, dropping the widgets from the pull down menu to the canvas and wiring them together.



Conclusions

Boeing Air Traffic Management

The screenshot displays the 'AirportSearch' application interface. At the top, there is a navigation bar with 'Lotus Mashups', 'AirportSearch', and user options 'admin | Help | Logout'. Below this is a search form titled 'Boeing Search Form' with fields for 'City' (Washington), 'State' (DC), 'Radius (in miles)' (20), and 'Min. Runway Length' (5000). A 'Search' button and a checked 'Show all airports' option are also present.

To the right of the search form is an 'OpenStreetMap' showing a map of the Washington, DC area with several airport icons. Below the map is an 'Airport Details' section with a table showing details for Ronald Reagan Washington National Airport (KDCA).

Below the search form is an 'Open Airports' section with a table listing nearby airports. Below the map is a 'Runway Details' section with a table showing runway information for the selected airport. Below that is a 'NOTAM Messages' section with a table of NOTAMs. At the bottom are 'RUC Messages' and 'METAR Messages' sections, each with a table of relevant data.

Airport_Name	ICAO	Airp
RONALD REAGAN WASHINGTON NATL	KDCA	US
ANDREWS AFB	KADW	US
DAVISON AAF	KDAA	US

Airport_Name	ICAO	Latitude	Longitude	Elevation
RONALD REAGAN WASHINGTON NATL	KDCA	38.852083	-77.037722	15.0

Length	Width	Surface_type
4805.0	150.0	ASP
5189.0	150.0	ASP
6869.0	150.0	ASP

Source_id	Account_id	Area_description
F	FDC	KDCA, WASHINGTON/NATL, DC, US , (385100N , 0770159W)
F	FDC	KDCA, WASHINGTON/NATL, DC, US , (385100N , 0770159W)
F	FDC	KDCA, WASHINGTON/NATL, DC, US , (385100N , 0770159W)

Icao	Last_Updated
KGAI	2009/05/12 17:35 at: Tue May 12 13:35:00 EDT 2009 wind: NNW 330 degrees at 7