ARCHITECTURE AND CAPABILITIES OF A DATA WAREHOUSE FOR ATM RESEARCH

Michelle M. Eshow, NASA Ames Research Center, Moffett Field, CA
Max Lui and Shubha Ranjan, Intrinsyx Corporation, Moffett Field, CA

Abstract

This paper describes the design, implementation, and use of a data warehouse that supports air traffic management (ATM) research at NASA's Ames Research Center. The data warehouse, dubbed Sherlock, has been in development since 2009 and is a crucial piece of the ATM research infrastructure used by Ames and its partners. Sherlock comprises several components, including a database, a web-based user interface, and supplementary services for query and visualization. The information stored includes raw data collected from the National Airspace System (NAS), parsed and processed data, derived data, and reports derived from pre-defined queries. The raw data include a variety of flight information from live streams of FAA operational systems, weather observations and forecasts, and NAS advisories and statistics. The modified data comprise parsed and merged data sources and metadata, enabling parameterized searches for data of interest. The derived data represent the results of research analyses deemed to be of significant interest to a wide cross-section of users. Sherlock is implemented on an Oracle 11g database, with supplemental services built on open-source packages and custom software. It contains over 20 TB of data spanning several years, and more data are added daily. It has supported several research studies, such as finding similar days in the NAS and predicting imposition of traffic flow management restrictions. Planned enhancements include integrated search across data sources and the capability for large-scale analytics.

Introduction

NASA Ames has a long history of productive research in ATM in the United States (US), particularly in the areas of decision support tools, automation, and simulation [1]. The success of Ames's ATM research has depended on access to actual data describing the NAS, including information about flights in the system, weather observations and forecasts, and advisory data issued by the FAA in response to current traffic, weather, or other conditions. NASA Ames is unique outside the FAA in that it receives live air traffic feeds from a variety of operational facilities, including Air Route Traffic Control Centers (Centers), Terminal Radar Approach Controls (TRACONs), and the Air Traffic Control System Command Center (ATCSCC). These data feeds drive the real-time decision support tools (DST) that Ames has developed [1]. They are also used for playback into the DST's, analysis, behavioral models, and as a baseline for developing simulation scenarios. Before the advent of Sherlock, the data feeds were archived to a file system on an as-requested basis. There was no consistent strategy for data archiving, nor was there a centralized storage location. The personnel recording the data supported each of the many recording requests from researchers individually. Knowledge about the data was transmitted informally, with little opportunity for discovery by others.

In 2008, a small team of engineers within the ATM group at Ames undertook the task of designing and creating a data warehouse to support ATM researchers at Ames and across NASA. The objectives at the time were to create a centralized repository of all relevant ATM data and to enable NASA researchers to query, view, and extract the data for their diverse purposes. As is standard practice, the warehouse would be nonvolatile and time variant, in that users would not be able to modify the data and new data would be added continuously. Data warehouses frequently serve to integrate data using a single data schema, but because of the challenging nature of integrating ATM data sources as diverse as aircraft position and storm locations, this aspect was taken into account in the design but not pursued at the outset of the project.

Sherlock's development proceeded in an iterative fashion, with each successive release giving meaningful new capabilities to users. The objective of the first release was modest: to store all ATM data in a consistent manner and to make it available to authorized users as 'raw' files, in day-long blocks, from both a file system and a web-based interface. Additionally, a small amount of metadata was specified and generated to quantify the completeness of the data. Users were able to search for data by date, airport or other facility, and type. They were able to view the completeness of the files and download acceptable ones for further use. Achieving these capabilities required re-engineering the existing data collection and archiving process to be fault-tolerant and as reliable as possible. It also required design of a directory hierarchy, file naming conventions, and a simple database schema. Finally, it required implementation of a web-based user interface.

The second release included storing some of the data sources according to a traditional database schema, which required that the data content be parsed and accessible through queries. The choice of which sources were parsed was determined by user needs and the difficulties of parsing the source data. The parsed sources help researchers find time periods of interest across the data. For example, a user could search for desired airport configurations for a range of dates; then among those dates search for highest arrival rates; and then search for the highest arrival delays. Particularly useful analyses created and carried out by individuals were moved into the warehouse to be updated continuously for use by anyone. Finally, software tools were made available to convert existing data into normalized databases and other data formats such as Keyhole Markup Language (KML) – the language used for such applications as Google Earth [2].

The third release addressed the acquisition and processing of additional data sources, including weather forecasts and a high-rate national air traffic feed. It also included the creation of services enabling query of weather and geographic data via software interfaces rather than by human input. A conceptual drawing of Sherlock is shown in Figure 1.

To date, Sherlock contains eight frequently accessed data source types in 'raw' format, eight types that are processed and placed in traditional database tables, one daily shared analysis, and three post-processing tools. Its data have been used in many research analyses and technical papers. The balance of this paper describes Sherlock's data, design, usage, and future plans.

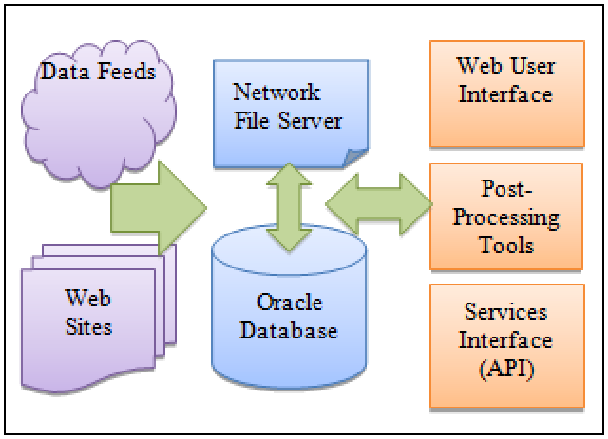


Figure 1. Overall Sherlock Architecture

Other ATM Data Warehousing Efforts

Other ATM-related organizations have built their own data archives, analysis tools, and reporting systems, but most are not widely accessible or do not have published descriptions. Perhaps the best-known ATM warehouse in the US is the Performance Data Analysis Reporting System (PDARS) [3]. At its lowest layer, PDARS collects flight plan and track data from nearly every facility in the US, including unique facilities in Alaska and Hawaii. PDARS also collects many weather products. It performs extensive data processing and provides over one thousand pre-defined reports daily to FAA users, from delay statistics to noise profiles. PDARS enables the FAA to assess overall NAS performance and make informed decisions about future operations. PDARS is an extremely powerful system, but it is not available for general NASA use due to security and design restrictions. It was, therefore, not suitable for NASA's research needs.

Sherlock Data Sources

Data Sources and Types

Data for Sherlock comes primarily from the FAA and the National Oceanic and Atmospheric Administration (NOAA). NASA has agreements with both organizations for proper use and dissemination of the data. Live flight plan and track data are not shared outside an isolated network; only archived data are available in the warehouse. Access to the archived data is limited to NASA and partners who have acquired NASA credentials. Only partners with legitimate needs and signed data usage agreements may access the archived data. The data sources are shown in Table 1 and discussed in more detail later in this paper.

Table 1. Sherlock Data Sources

Source Name	Description	Acquisition
ATCSCC Advisories [4]	Traffic management advisories issued by System Command Center, in HTML format	Retrieved from FAA website daily.
Airline Situation Display to Industry (ASDI) [5]	National air traffic, updated once per minute, in a compressed Extensible Markup Language (XML) feed	FAA product streamed over Virtual Private Network (VPN), recorded in one-minute-long files.
Air Route Traffic Control Center (Center)	Flight plan and track data from a Center computer, distributed through a gateway, binary format	Streamed over dedicated network from FAA. Recorded as one file per day per Center.
Corridor Integrated Weather Service (CIWS) [6]	Current and forecast precipitation and echo tops for continental US, in binary format	Retrieved from FAA site over VPN, multiple files every 2.5 minutes.
Center-TRACON Automation System (CTAS) [1]	Custom file generated at Ames, used to record merged Center and TRACON data, including automation data, in text format	Recorded continuously into one file per day per Center-TRACON combination.
Exelis Commercial Track Feed [7]	National air traffic feed, updated every five seconds or more, XML format	Streamed over VPN for six weeks total in 2013, recorded into one-minute-long files.
Meteorological Aerodrome Reports (METAR) [8]	Hourly surface weather observations from hundreds of airports in US, in text format	Retrieved from NOAA file transfer protocol (FTP) site hourly.
The Operations Network (OPSNET) [9]	FAA's aggregate system performance data for US, by airport and day, including delay by cause, in comma-separated value (CSV) format	Retrieved from FAA OPSNET website monthly.
Aircraft Reports [10, 11]	Pilot and aircraft reports of in-flight weather conditions across US, in text format	Retrieved from NOAA website daily.
Rapid Refresh (RR) Weather Forecast [12]	NOAA weather forecasts, including wind, temperature, and pressure, published in GRIdded Binary (GRIB) format	Retrieved from NOAA FTP site hourly.
Terminal Aerodrome Forecast (TAF) [13]	Forecasts of airport weather, published every six hours for many airports, in text format	Retrieved from NOAA FTP site hourly.
Time-based Flow Management (TBFM) metering information [14]	FAA’s daily summaries of metering usage and related data extracted from TBFM software, in CSV and HTML format	Delivered from FAA analysis system once per day.
TRACON	Flight plan and track data from TRACON computers, through a gateway, in binary format	Streamed over dedicated network, one stream per TRACON.

Data Management

Key to the success of any data warehouse is reliable and robust collection of data on a continuous basis. For each data source identified in Table 1, Sherlock has an automated process to connect to the source, receive or retrieve the data, and write it to a central file system. In some cases there is a dedicated or private virtual network connection receiving a continuous feed, as is true for the Center and TRACON flight data. In other cases, especially for weather data, polling programs connect to a remote FTP server at defined intervals and look for new files. In all cases, the retrieval software runs on the Ames ATM computer network to bring the data across Linux servers to a network file system (NFS). Separate programs monitor the data collection of each source to detect missing data, such as a break in a continuous stream or a missing file on an FTP site. When a break in the data is detected, the scripts send email to appropriate personnel and catalog the missing data. At the end of each day, other scripts acquire missing files from remote servers, for data sources that have some level of archiving. Finally, scripts run overnight to rename the files according to Sherlock's naming conventions and move the data to a permanent place in the file system's directory hierarchy. Because of the monitoring and recovery scripts noted above, there is typically a one-day delay between the data arriving at Ames and it being available through the web application.

Data Extraction, Transformation, and Loading (ETL)

After the data are collected, they must be loaded into or referenced by the Oracle database component of the warehouse. Oracle was selected as the database application because its table partitioning feature allows fast queries across large amounts of data, and because it allows many database operations to run in parallel (e.g., data inserts and index creation). The Oracle Developer Data Modeler application is used to model and maintain the database tables.

The data loading step is typically called "extraction, transformation, and loading", or ETL. In most cases, Sherlock's ETL process is implemented using Pentaho Data Integration (PDI), an open source application written in Java [15]. The ETL software generated by PDI is metadata driven. A developer uses PDI to construct the software by connecting multiple plug-and-play steps, using the PDI graphical user interface. PDI can use parallel data processing to execute the ETL software, which is ideal when running on a computer with multiple CPU cores. Figure 2 shows a graphical presentation of part of the ETL process to decode and load METAR data into a database, per the methods described in [16].

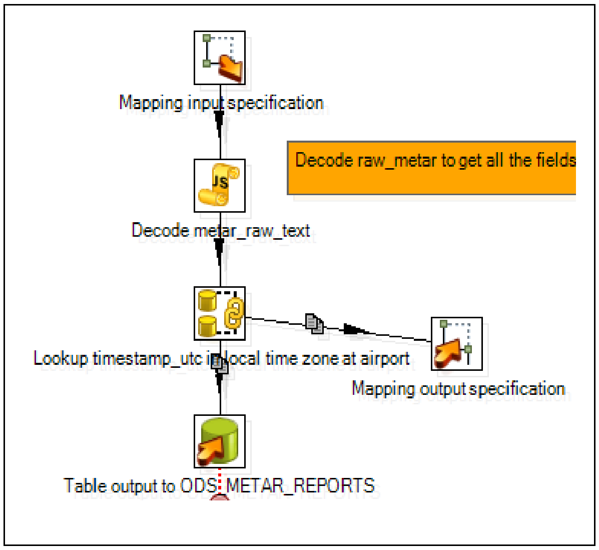

Mapping input specification
Decode raw_metar to get all the fields
Decode metar_raw_text
Lookup timestamp_utc in local time zone at airport
Mapping output specification
Table output to ODS.METAR_REPORTS

Figure 2. ETL Process for METAR in PDI

The ETL software first loads data into a staging database. The input data source may be a file, another database, a web page, or a web service call. The data in the staging database may be further transformed into a Star Schema form [17], a modeling approach optimized for ad-hoc user queries of large amounts of data. Figure 3 shows the daily aggregated Star Schema form for METAR. The dimension tables D_WX_PHENOMENA, D_DATE, and D_AIRPORT store descriptive data about weather phenomena, dates, and airports, respectively. The fact table F_METAR_REPORT_DAILY_AGG stores the daily aggregated facts (surface-weather summary observed at an airport on a given date).

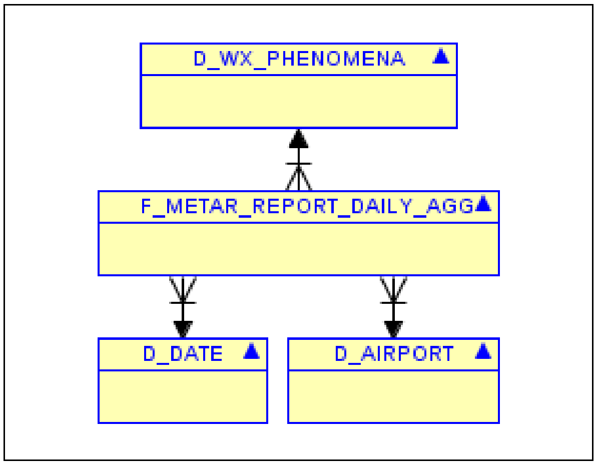
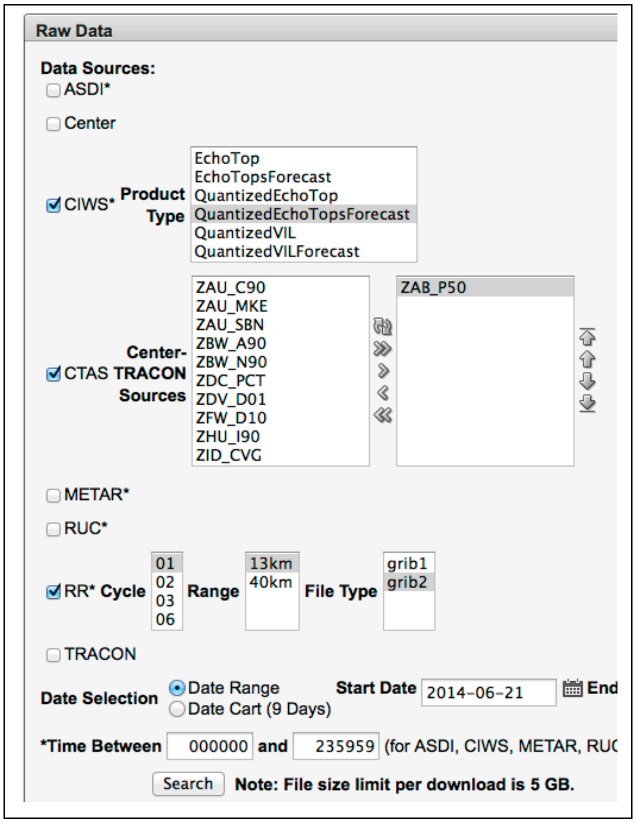


Figure 3. Star Schema for METAR

Data Access in Sherlock

All data in the database are accessible by Sherlock's web application, which is built upon Oracle Application Express (APEX). APEX was selected because it promised rapid web application development, and its end-user reporting capability met early requirements for ad-hoc filtering, sorting, grouping, and charting.

Figure 4 illustrates part of the main search page, where users look for data by source and dates. In this example, the user is interested in the CIWS, CTAS, and RR data sources. The user has selected sub-sets of each of those sources, as indicated by the selections highlighted in grey.



Raw Data

Data Sources:

□ ASDI*

□ Center

☑ CIWS* Product Type: EchoTop
EchoTopsForecast
QuantizedEchoTop
QuantizedEchoTopsForecast
QuantizedVIL
QuantizedVILForecast

☑ CTAS Center-TRACON Sources: ZAU_C90
ZAU_MKE
ZAU_SBN
ZBW_A90
ZBW_N90
ZDC_PCT
ZDV_D01
ZFW_D10
ZHU_I90
ZID_CVG — ZAB_P50

□ METAR*

□ RUC*

☑ RR* Cycle 01 02 03 06 Range 13km 40km File Type grib1 grib2

□ TRACON

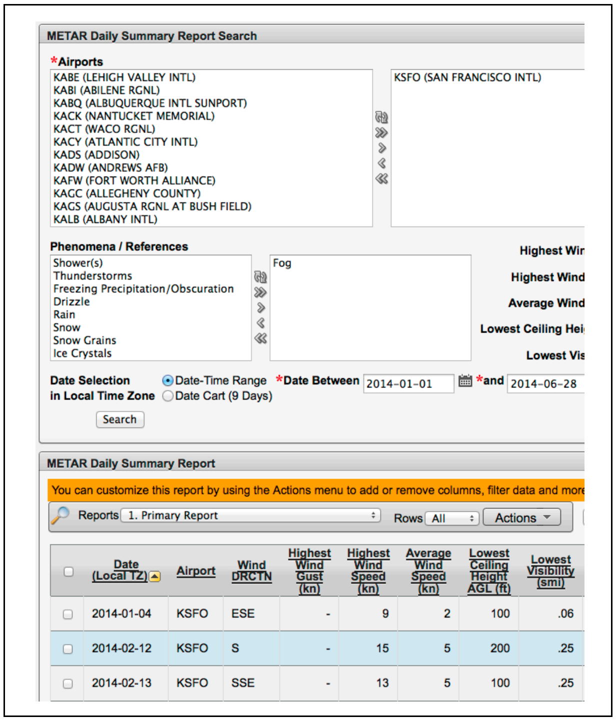
Date Selection ● Date Range ○ Date Cart (9 Days) Start Date 2014-06-21 End

*Time Between 000000 and 235959 (for ASDI, CIWS, METAR, RUC

Search Note: File size limit per download is 5 GB.

Figure 4. Same Query Across Sources

Figure 5 shows the query page for METAR summary data. Since METAR is a parsed source, the content of the weather reports is available for query. The APEX interface for METAR was designed for search by airports, weather conditions, and dates. In this example, the user has searched for Fog conditions at San Francisco International Airport from January 1st, 2014 to June 28th, 2014. The columns displayed in this example include the 'worst' conditions for that day, including the lowest visibility. Users may select or deselect every parameter in the data for display. In addition, the users may perform additional formatting, filtering, and computation on the results. Users may download the resulting data tables in CSV or HTML format. APEX also provides a variety of charting options



METAR Daily Summary Report Search

*Airports

KABE (LEHIGH VALLEY INTL)
KABI (ABILENE RGNL)
KABQ (ALBUQUERQUE INTL SUNPORT)
KACK (NANTUCKET MEMORIAL)
KACT (WACO RGNL)
KACY (ATLANTIC CITY INTL)
KADS (ADDISON)
KADW (ANDREWS AFB)
KAFW (FORT WORTH ALLIANCE)
KAGC (ALLEGHENY COUNTY)
KAGS (AUGUSTA RGNL AT BUSH FIELD)
KALB (ALBANY INTL)

KSFO (SAN FRANCISCO INTL)

Phenomena / References

Shower(s)
Thunderstorms
Freezing Precipitation/Obscuration
Drizzle
Rain
Snow
Snow Grains
Ice Crystals

Fog

Highest Wir
Highest Wind
Average Wind
Lowest Ceiling Hei
Lowest Vis

Date Selection in Local Time Zone ● Date-Time Range ○ Date Cart (9 Days) *Date Between 2014-01-01 *and 2014-06-28

METAR Daily Summary Report

You can customize this report by using the Actions menu to add or remove columns, filter data and more

Reports 1. Primary Report Rows All

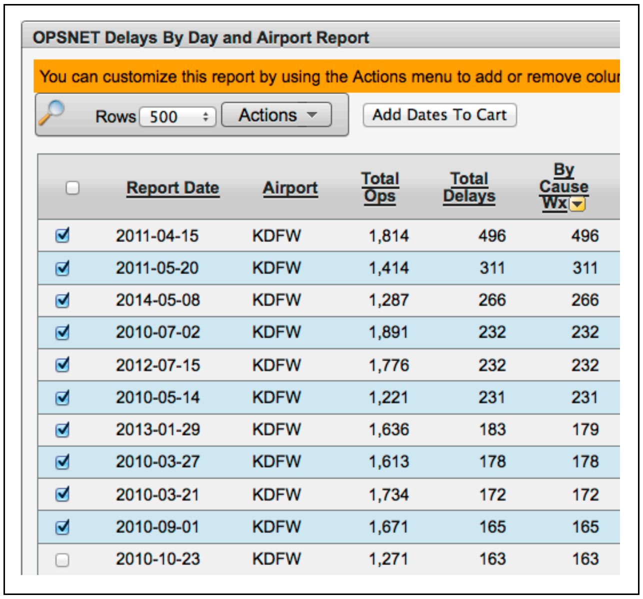
□	Date (Local TZ)	Airport	Wind DRCTN	Highest Wind Gust (kn)	Highest Wind Speed (kn)	Average Wind Speed (kn)	Lowest Ceiling Height AGL (ft)	Lowest Visibility (smi)
□	2014-01-04	KSFO	ESE	-	9	2	100	.06
□	2014-02-12	KSFO	S	-	15	5	200	.25
□	2014-02-13	KSFO	SSE	-	13	5	100	.25

Figure 5. User Query of METAR Data

Date Cart for Date-Related Search

One of the main use cases for Sherlock is to find data with traffic or weather conditions of interest. The first search criterion is a date range, and this can be specified over the entire coverage period of data for a source. While examining sources across a range of dates, a user may find discrete dates with conditions of interest, such as days of high delay due to severe weather and days of low delays in the same geographical region. These insights are gained by successive searches across various dates. To help the users save dates of interest as they find them, the interface provides a Date Cart that is similar to a shopping cart used in many online commerce web sites. The user may add a range of dates or individual dates to the date cart from any search result. The dates in the cart can be used subsequently as a search criterion throughout the web application.

Figure 6 shows the results of searching OPSNET data for the most weather-impacted days at Dallas-Ft. Worth Airport (DFW), between January 1st, 2010 and May 31st, 2014. The user has selected the top ten dates for further use. After the user clicks "Add Dates to Cart," those dates are available for all other searches. Figure 7 shows use of the Date Cart as a search criterion on another page.



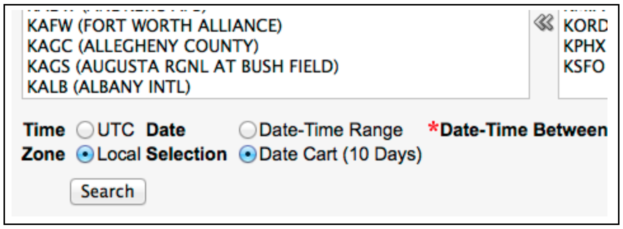
OPSNET Delays By Day and Airport Report

You can customize this report by using the Actions menu to add or remove colu

Rows 500

□	Report Date	Airport	Total Ops	Total Delays	By Cause Wx
☑	2011-04-15	KDFW	1,814	496	496
☑	2011-05-20	KDFW	1,414	311	311
☑	2014-05-08	KDFW	1,287	266	266
☑	2010-07-02	KDFW	1,891	232	232
☑	2012-07-15	KDFW	1,776	232	232
☑	2010-05-14	KDFW	1,221	231	231
☑	2013-01-29	KDFW	1,636	183	179
☑	2010-03-27	KDFW	1,613	178	178
☑	2010-03-21	KDFW	1,734	172	172
☑	2010-09-01	KDFW	1,671	165	165
□	2010-10-23	KDFW	1,271	163	163

Figure 6. Adding Weather-Impacted Dates to Cart



KAFW (FORT WORTH ALLIANCE)
KAGC (ALLEGHENY COUNTY)
KAGS (AUGUSTA RGNL AT BUSH FIELD)
KALB (ALBANY INTL)

KORD
KPHX
KSFO

Time Zone ○ UTC ● Local

Date Selection ○ Date-Time Range ● Date Cart (10 Days) *Date-Time Between

Figure 7. Date Cart as a Search Parameter

File Metadata and File Download

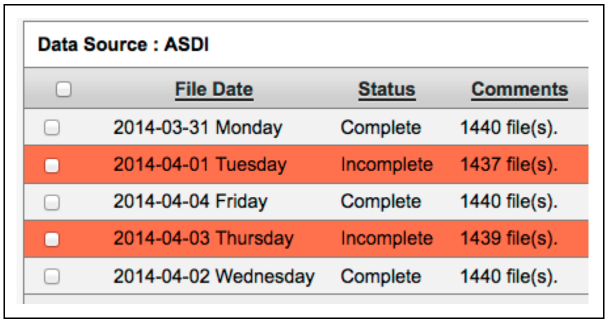
As mentioned previously, the recording of streaming data is subject to occasional outages. For example, the network connection can be disrupted upstream, or the servers may be shut down for unplanned maintenance. Because the data are not operationally critical, there is currently no failover recording capability and no contingency for recovering any lost live-stream data. To aid a user in assessing the value of data for a given day, the database stores metadata regarding each file's availability and completeness. These metadata are presented with search results for the air traffic and weather data sources. Below are the metrics of completeness for the data sources:

- ASDI: 1440 one-minute files per day
- CIWS: 576 sets of files per day (one set of 4 files every 2.5 mins)
- CTAS, Center, TRACON: Internal recording time stamps spanning 24 hours

- METAR: 24 files per day
- RR: a set of five files per hour

A user can conduct a ‘completeness search’ over a range of dates or across the dates stored in the date cart. It should be noted that data integrity is not currently verified; that is an area for future work.

Figure 8 shows a sample data search query result in which there are some short (1-3 minute) lapses in ASDI data. These are highlighted in red. If the data lapses occurred during non-peak air traffic, a user may still find the data usable. A separate window enables the user to see what time periods are missing. Knowing more about the missing data allows the user to make an informed decision about whether to use those days or choose others.



Data Source : ASDI

☐	File Date	Status	Comments
☐	2014-03-31 Monday	Complete	1440 file(s).
☐	2014-04-01 Tuesday	Incomplete	1437 file(s).
☐	2014-04-04 Friday	Complete	1440 file(s).
☐	2014-04-03 Thursday	Incomplete	1439 file(s).
☐	2014-04-02 Wednesday	Complete	1440 file(s).

Figure 8. Data Completeness on Search

To download files of interest, the user selects the check-box beside the corresponding file(s) then clicks the "Download Selected Files" button. The web application will then generate an archive of all selected files and present it to be saved from the user's browser.

Data Stored in Database Tables

The following sections describe the data sources that are stored in database tables rather than solely as data files in their original format. The user may query within each source using the built-in capabilities of the APEX interface. The user may also search across sources by constructing complex queries, but this requires knowledge of the database schema and use of Structured Query Language (SQL).

OPSNET

OPSNET [9] provides aggregated data about NAS air traffic operations and delays. Sherlock stores the OPSNET data from the Aviation System Performance Metrics – 77 airports [18] beginning January 1st, 2007. Figure 9 shows the three days with the highest total delays at Newark Liberty Airport between May 1st, 2013 and April 30th, 2014. Users can retrieve this data for analysis or insert these dates into their Date Cart for wider searches. OPSNET data are also used in computing the Weather Impacted Traffic Index as described later in this paper.

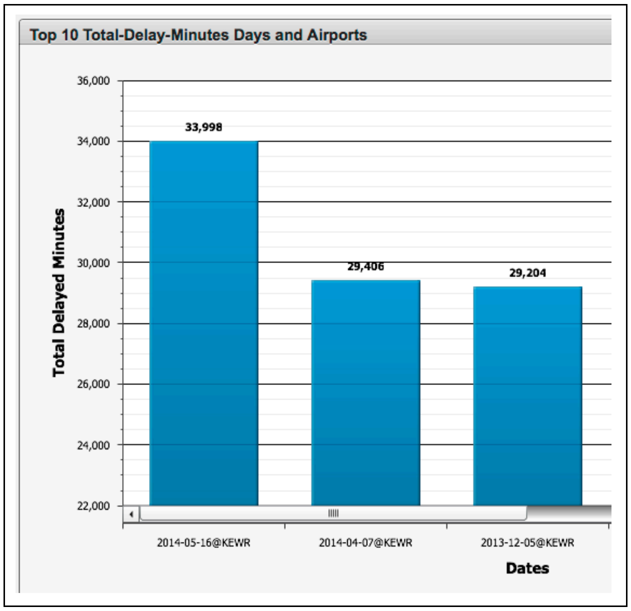


Figure 9. Days of Highest Delay at Newark Airport

METAR

A METAR [8] report contains weather information observed by an automated or manual reporting station at an airport, and covers the immediate vicinity of that airport. A regular report is created every hour, and a special report is generated when there is a significant change in conditions. Each report has two sections of text: a Body and Remarks. Sherlock collects the METAR reports from a NOAA FTP site, with the Weather Underground website as an alternative source. METAR reports from 267 airports are decoded and stored on a daily basis, using the methods described in [16].

Time-Based Flow Management Information

Operational TBFM data are extracted from all 20 Centers and the TRACONs that run the TBFM software [14]. These data help researchers identify traffic conditions of interest. They may query the following TBFM data:

- Airport flow configuration, including runway landing direction and active runways.
- Airport landing summary, including total aircraft landed, average landing per hour, and average landing per 10-minute or 15-minute intervals.
- Metering usage, including when arrival metering started and ended.

Weather Impacted Traffic Index

Based on several of its sources, Sherlock computes an hourly Weather-Impacted Traffic Index (WITI) [19] for each Center in the NAS. WITI is computed because of its wide utility for classifying traffic conditions. This processing uses ASDI and CIWS files directly. OPSNET data are used to choose ASDI low-weather-impact days for comparison with the subject day's weather. WITI is a measure of the effects weather had on en-route traffic flow over an hour or a day. An Hourly WITI having a value of one is defined as one aircraft-minute of delay that was incurred due to a level of precipitation that a pilot would likely avoid using flight path changes.

ATCSCC Advisory

Advisories issued by the FAA's ATCSCC may impact air traffic, especially when the advisory is a Ground Delay Program, Airspace Flow Program, Ground Stop, or Reroute. The advisory data are parsed and stored to allow users to fully query all of the associated parameters (e.g., impacting condition such as thunderstorms, and affected facilities such as airports or Centers). The advisory data source is also key to understanding air traffic flows when playing back historical data.

ASDI Track Analysis

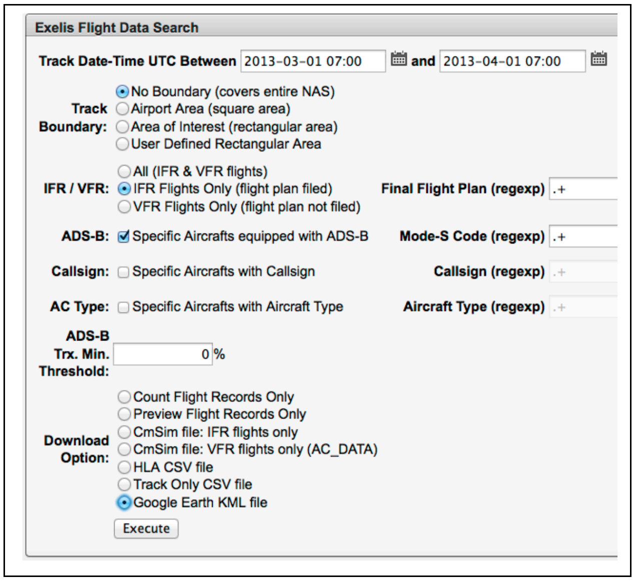
The ASDI data feed is streamed and recorded in Sherlock in its native format, in one-minute file intervals. FAA creates the ASDI feed from many track sources, including Center and TRACON facilities. Each day of data contains about five million tracks. During ETL processing, all of the tracks are aggregated based on their track sources to compute the number of tracks received every hour. Users may then query for and rank hourly traffic levels at selected Centers or TRACONS. Again, the purpose of this processing is to help researchers find traffic conditions of interest for further analysis.

Aircraft Reports

Aircraft Reports are made up of Air Reports (AIREP) and Pilot Reports (PIREP) [10, 11]. Both are available from the National Weather Service ADDS Data-Services Text-Data-Server website [20]. These reports come from pilots and describe their flight's position and the presence or absence of adverse weather conditions such as turbulence, icing, visibility, and wind. ETL scripts parse the message text before storing the parsed data into the database. Sherlock users can query and filter historical aircraft reports to assess the weather conditions encountered by en-route flights, and to compare measurements provided by weather models against those from the pilot reports.

Exelis Flight Data

Exelis NextGen Surveillance Data Services [7] provides a commercial real-time feed of all transponding aircraft in the NAS as well as ground tracks for many airports. The track data comes from FAA facilities and from Automatic Dependent Surveillance-Broadcast (ADS-B) signals, where available. To support creation of simulation scenarios for unmanned aircraft system research, Ames partnered with Exelis to purchase a 50-day subscription to the data stream. Software at Ames recorded the feed in one-minute-long XML files. The team wrote extensive processing code to parse, de-duplicate and clean the data. As a result, 9.6 million flight information records and 3.4 billion track records are available for query, charting and download from Sherlock's interface. To allow existing NASA tools and applications to read the Exelis data, the team developed software to support several export formats: CTAS Cmsim (to be described later in this paper), High Level Architecture (HLA) Comma-Separated Values (CSV) [21], and KML [2]. Figure 10 shows how a user can define data export criteria based on spatial and time boundary of tracks, call-signs, ADS-B transponder codes, aircraft types, etc. Figure 11 shows the playback of a few tracks and track histories at DFW using Google Earth; this allows a quick visualization of the nature of the flight and track data in an export file.



Exelis Flight Data Search

Track Date-Time UTC Between 2013-03-01 07:00 **and** 2013-04-01 07:00

Track Boundary:
● No Boundary (covers entire NAS)
○ Airport Area (square area)
○ Area of Interest (rectangular area)
○ User Defined Rectangular Area

IFR / VFR:
○ All (IFR & VFR flights)
● IFR Flights Only (flight plan filed)
○ VFR Flights Only (flight plan not filed)

Final Flight Plan (regexp) .+

ADS-B: ☑ Specific Aircrafts equipped with ADS-B

Mode-S Code (regexp) .+

Callsign: □ Specific Aircrafts with Callsign

Callsign (regexp) .+

AC Type: □ Specific Aircrafts with Aircraft Type

Aircraft Type (regexp) .+

ADS-B Trx. Min. Threshold: 0 %

Download Option:
○ Count Flight Records Only
○ Preview Flight Records Only
○ CmSim file: IFR flights only
○ CmSim file: VFR flights only (AC_DATA)
○ HLA CSV file
○ Track Only CSV file
● Google Earth KML file

Figure 10. Search Page for Exelis Track Data

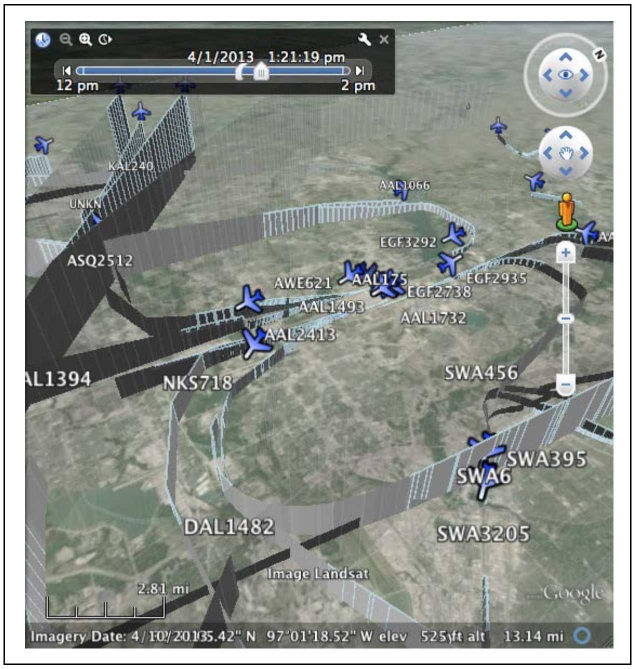


Figure 11. Exelis Tracks on Google Earth

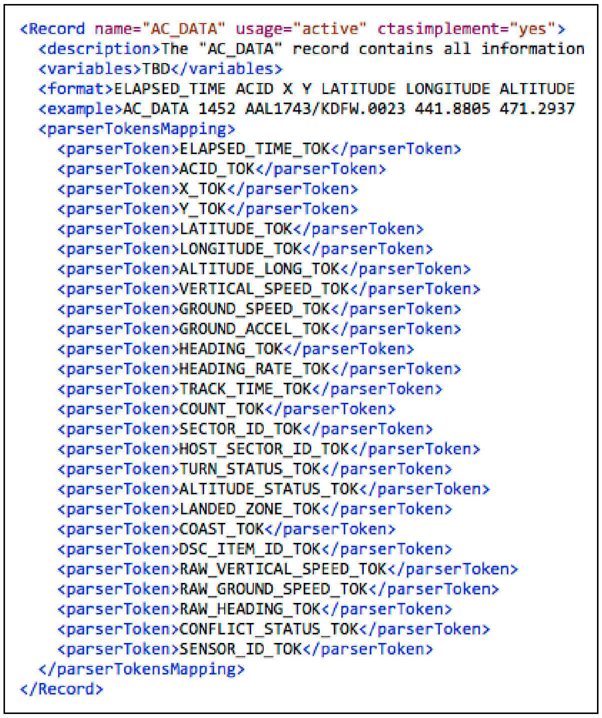
CTAS Data File Parsing

NASA's Center-TRACON Automation System (CTAS) [1] is a large software baseline that embodies a collection of real-time, controller-centered ATM decision support tools. The currently fielded TBFM system began as a component of CTAS in the 1990's called the Traffic Management Advisor [22]. To support development, a customized data-recording format was created to store data from CTAS for analysis and playback. This data type is called the "Cmsim" file. The Cmsim file is text-based and contains 40 different record types reflecting both flight data that comes into CTAS from FAA systems, and data internally generated by CTAS based on its predictions. The records have structured as well as unstructured text fields. Each Cmsim file recorded for Sherlock covers a 24-hour period for a given combination of one Center and one TRACON. The types of records include track and flight plan data, estimated and scheduled times of arrival, assigned meter fix and runway, and pointers to any weather files that were in use during the recording. A 24-hour Cmsim file holds 4 to 10 million unique records, depending on the connected ATC facilities and level of traffic.

Because the Cmsim file format is used so widely by Ames researchers, the Sherlock team wrote a program to read and process the records into other formats amenable to analysis and visualization. The Java-based Cmsim Parser tool is able to quickly read a Cmsim file of up to 15 million records into local memory, then process the data and export it to many different formats, including: a subsetted Cmsim file (described below), a MATLAB file, a CSV file, or database tables readable by all common database applications. The database export also provides a means to merge several days of data together and query across them.

The Cmsim Parser's web interface enables the user to specify complex selection and filtering criteria for the data, as well as choose among output formats. The program can be used in a standalone, interface-driven mode, by terminal command line, or by other programs via an application programming interface (API). The uniquely powerful aspect of the Parser is that it is metadata driven; its knowledge of the Cmsim file format is determined completely by an external XML representation of the records, rather than by any record-specific coding within the program. A truncated sample of the XML specification is presented in Figure 12.

In addition to parsing the data for analysis, a popular use of the program is to generate a "mini Cmsim" file. Users may pick a subset of flights from the original file and create a smaller one that retains all of the required syntax of the original. They may then play back the file into CTAS, to more easily explore system behavior for those few flights.



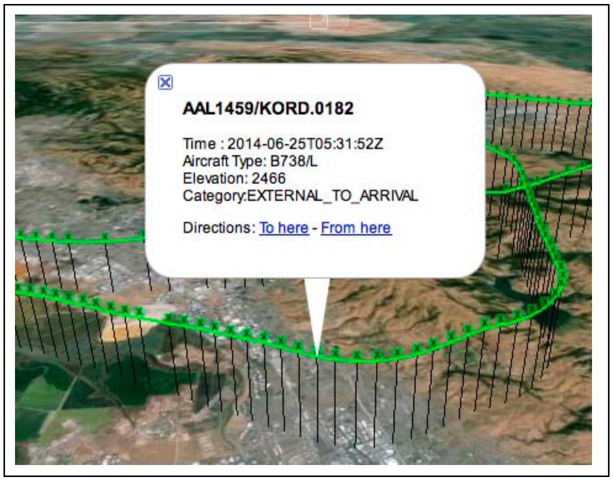
```
<Record name="AC_DATA" usage="active" ctasimplement="yes">
  <description>The "AC_DATA" record contains all information
  <variables>TBD</variables>
  <format>ELAPSED_TIME ACID X Y LATITUDE LONGITUDE ALTITUDE
  <example>AC_DATA 1452 AAL1743/KDFW.0023 441.8805 471.2937
  <parserTokensMapping>
    <parserToken>ELAPSED_TIME_TOK</parserToken>
    <parserToken>ACID_TOK</parserToken>
    <parserToken>X_TOK</parserToken>
    <parserToken>Y_TOK</parserToken>
    <parserToken>LATITUDE_TOK</parserToken>
    <parserToken>LONGITUDE_TOK</parserToken>
    <parserToken>ALTITUDE_LONG_TOK</parserToken>
    <parserToken>VERTICAL_SPEED_TOK</parserToken>
    <parserToken>GROUND_SPEED_TOK</parserToken>
    <parserToken>GROUND_ACCEL_TOK</parserToken>
    <parserToken>HEADING_TOK</parserToken>
    <parserToken>HEADING_RATE_TOK</parserToken>
    <parserToken>TRACK_TIME_TOK</parserToken>
    <parserToken>COUNT_TOK</parserToken>
    <parserToken>SECTOR_ID_TOK</parserToken>
    <parserToken>HOST_SECTOR_ID_TOK</parserToken>
    <parserToken>TURN_STATUS_TOK</parserToken>
    <parserToken>ALTITUDE_STATUS_TOK</parserToken>
    <parserToken>LANDED_ZONE_TOK</parserToken>
    <parserToken>COAST_TOK</parserToken>
    <parserToken>DSC_ITEM_ID_TOK</parserToken>
    <parserToken>RAW_VERTICAL_SPEED_TOK</parserToken>
    <parserToken>RAW_GROUND_SPEED_TOK</parserToken>
    <parserToken>RAW_HEADING_TOK</parserToken>
    <parserToken>CONFLICT_STATUS_TOK</parserToken>
    <parserToken>SENSOR_ID_TOK</parserToken>
  </parserTokensMapping>
</Record>
```

Figure 12. Sample XML Representation

Track Visualizer

Building on the Cmsim Parser, the Track Visualizer is a web application that generates KML output to display flight tracks on Google Earth or any other geospatially-enabled browser. It uses the Cmsim Parser API to parse a user-selected Cmsim file. The visualizer is built using HTML5 [23] and operates on any computer or browser supported by the Google Earth plug-in. The Track Visualizer connects to the Oracle database, so users can select any stored Cmsim file. Users may also upload their own Cmsim file.

The Track Visualizer provides several KML generation options, including display of aircraft call sign and color-coding of flight category (arrival, departure, etc.). Users may choose any flights of interest for KML processing. Their chosen flights then display in the browser using the Google Earth plugin. The generated track files can also be downloaded for further analysis, and airspace boundaries are provided as a layer. Figure 13 shows an example of a full playback of two sets of track data on Google Earth. The KML for each track point includes information about the flight, and this is visible when the user clicks on the point.



AAL1459/KORD.0182

Time : 2014-06-25T05:31:52Z
Aircraft Type: B738/L
Elevation: 2466
Category:EXTERNAL_TO_ARRIVAL

Directions: To here - From here

Figure 13. Example of Rendering of KML Output

Geospatial Service

To support geospatially oriented queries, visualization, and analysis, the team deployed a geospatial service using open-source tools. The service's layered architecture is illustrated in Figure 14. The PostgreSQL open-source database [24] is augmented with the PostGIS [25] extension to provide a variety of spatial features compliant with OpenGIS [26] standards. This integration comprises the GeoServer open-source server [27]. The service's database is populated with a sub-set of the airspace description data stored in Sherlock, including polygon representations of Center and sector boundaries, line descriptions of airways, and point descriptions of fixes, airports, and runways. In addition, the PostGIS database stores a polygonal representation of CIWS storm data that considers pilot interaction with the airspace around storm cells called the Convective Weather Avoidance Model (CWAM) [28]. These data are updated regularly to capture any changes from the file system.

The service provides multiple forms of interaction. Programs can connect to the database and run queries to answer such questions as "which sector(s) contain this storm polygon" or "what is the intersection point between these two airways". These queries are made possible by the function, operator, and index extensions of PostGIS.

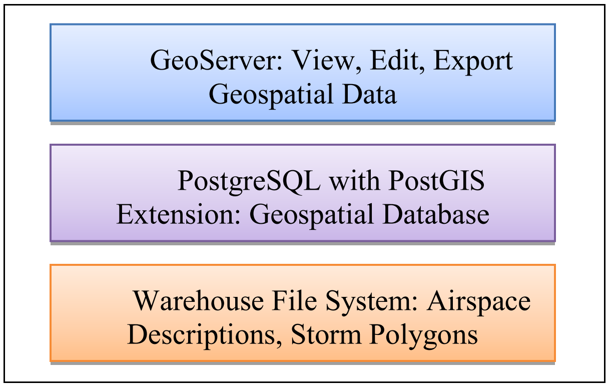

GeoServer: View, Edit, Export Geospatial Data
PostgreSQL with PostGIS Extension: Geospatial Database
Warehouse File System: Airspace Descriptions, Storm Polygons

Figure 14. Geospatial Service Layered Architecture

The GeoServer enables users to browse the various stored data types in both visual and text forms, and download the data in a preferred format. One example use case is to download CWAM data as a KML layer and then overlay it with flight tracks on Google Earth. The GeoServer provides airspace definition data in 20 possible formats including CSV, KML, and images. This is done via a complex query known as a Web Map Service (WMS) query. Figure 15 illustrates the results of a simple query of airspace fixes around Los Angeles International Airport (LAX), on Google Earth.

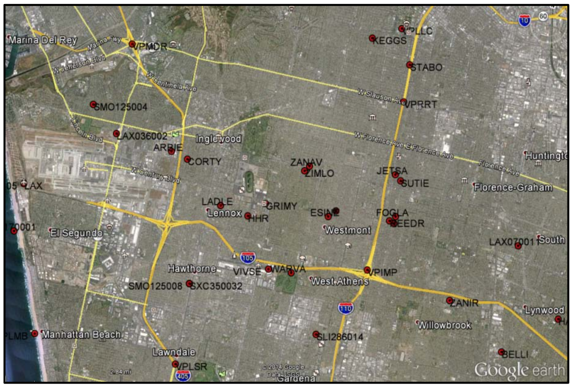


Figure 15. Fixes around LAX, retrieved from GeoServer, displayed on Google Earth

Weather Server

Weather conditions are often critical to ATM analysis. Sherlock stores large amounts of binary weather forecast data, including Rapid Refresh and CIWS. At times, researchers want to find particular wind or storm conditions, which requires delving into the complex data structures within the files. Since the files are in standard gridded formats, open source tools exist to process and query them. The team has implemented a weather server using the THREDDS Data Server (TDS) open source software [29] to enable this type of data query.

The THREDDS server installation comprises multiple virtual machines hosted in the ATM Linux network. It runs on a load-balanced, J2EE- compliant web server. It uses four nodes for performance and scalability to support the volume of data that comprise the weather datasets.

TDS allows users to find weather datasets that are pertinent to their specific research needs, access the data, and use it without necessarily downloading the entire file to their local system. TDS has integrated viewers, both web-based and Java-based, for viewing the data in various gridded data formats. TDS can serve the contents of the datasets, in addition to providing catalogs of all the files and metadata for them. TDS uses the Unidata [30] Common Data Model (CDM) to read datasets in various formats; it serves them through OPeNDAP (Open-source Project for a Network Data Access Protocol) [31].

The server can be queried with hyper-text transfer protocol (HTTP) calls, which can become complex and are intended to be generated as the user navigates the system or by an external application. One actual use case was to search for high cross-winds over a fix along an arrival route. Using the TDS-defined URL specification, a simple utility was created to iterate over a time/date and create a table of winds over the fix by time. The researcher quickly identified a Rapid Refresh weather file containing an acceptable wind magnitude and direction over the fix.

The stored weather data are served using standard web services including the Open Geospatial Consortium (OGC) Web Coverage Service (WCS) and the WMS. The server can provide subsets of the digital data in various formats using NetCDF sub-setting, which is another web service. For example, a user may specify a bounding box to retrieve storm data over a certain geographic region. Similarly the user may specify a subset of parameters to be retrieved, such as freezing rain or wind vectors at a given location and time. The sub-set preserves the original resolution.

The weather server can generate visualizations in all standard image formats or as a KML layer.

Several open-source web-based tools have been added to allow the user to specify distinct layers, spatial reference system, geographic area, and other parameters for the returned map format. Figure 16 shows a wind vector visualization using the 'wind at maximum wind level' parameter from a Rapid Refresh file. The wind barbs indicate direction, while the color indicates speed.

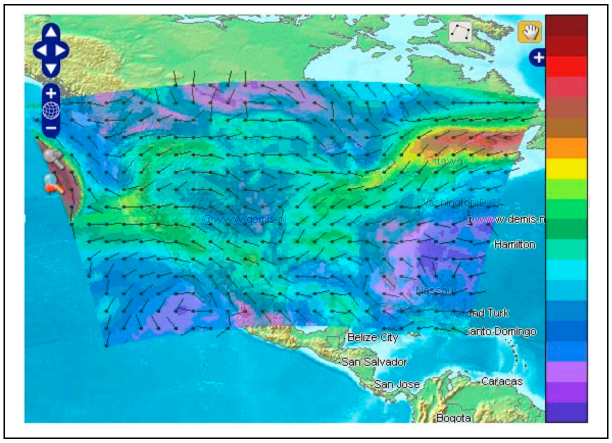


Figure 16. Wind Vector Image from WMS Query

Sherlock Usage Examples

In addition to daily routine usage, NASA and affiliated researchers have used Sherlock in a variety of studies leading to published works. This section summarizes a selection of such use cases.

A series of studies have applied data mining techniques to identify past occurrences of similar days in the NAS, to enhance FAA Traffic Flow Management (TFM) decision-making when such days occur in the future. Grabbe et al. [32] mined data about FAA-imposed Ground Delay Programs (GDPs) to find clusters across GDP locations and causes. This work used the METAR and ATCSCC Advisories sources from Sherlock. Another effort [33] used the derived WITI data product to identify six dominant weather patterns across the US, while the ATCSCC Advisories source helped to correlate re-route advisories with those weather patterns.

Researchers extended this work on analyzing similar days to identify clusters of hours for which the probability of imposing a GDP was similar, for the Chicago O'Hare and Newark Liberty airports [34]. This study used the ATCSCC Advisories and CIWS data sources.

Bloem et al. [35] created behavioral cloning and inverse reinforcement learning models to predict hourly GDPs at San Francisco and Newark Liberty airports. This work used Sherlock's METAR and TAF data.

As a final example, all of the development of the real-time Dynamic Weather Routing Tool [36] has depended heavily on Sherlock, especially to retrieve historical CIWS and Cmsim data for playback into the software, for verification, and for examination of situations reported by airline users.

Future Directions

Sherlock is an important infrastructural element for NASA's ATM research community. There are several areas of development that would increase its utility. First, the flight plan and track data should be processed or at least correlated into end-to-end flight records for each flight. This is a challenging task that requires merging data from many facilities into a single set of records per aircraft, while accounting for overlapping coverage areas, properly associating aircraft across facilities, and linking a particular track to the flight plan in effect at the time. ASDI data can't be used for all research purposes because of its inherent data filtering and because its one-minute update rate is not sufficiently frequent for detailed analyses. To be even more useful, metadata should be computed for each flight's track data, for example, actual landing runway, average in-trail spacing, and arrival meter fix crossing time. In addition, correlation of track data with automation system computations, such as trajectory predictions, has been identified as a useful capability.

The second planned enhancement is to provide more comprehensive data integration across Sherlock's multiple data sources. Toward this goal, a semantic model of ATM-related concepts has been constructed to integrate data from selected data sources, using an ontological approach [37]. Once completed, this work will enable cross-source queries to return results based on inferred relationships that are not explicit in the data.

The third planned enhancement is to add the capability for Big Data analytics of the type described by Mayer-Schonberger and Cukier [38]. With a functioning Big Data capability, sophisticated data mining and machine learning algorithms could be executed directly in Sherlock, without the need for researchers to download and process the data on their own computers. Efforts are underway to design this capability, and a prototype has been developed. Implementing Big Data depends on successful creation of the parsed end-to-end track data described as the first enhancement.

References

[1] Schroeder, J. A., 2009, "A Perspective on NASA Ames Air Traffic Management Research," AIAA-2009-7054, Hilton Head, SC, AIAA Aviation, Technology, Integration, and Operations (ATIO) Conference and Aircraft Noise and Emissions Reduction Symposium (ANERS).

[2] KML Standard, refer to: http://www.opengeospatial.org/standards/kml/

[3] Browder, Jeff, R. Gutterud, and J. Schade, 2010, “Performance Data Analysis Reporting System (PDARS) – A Valuable Addition to FAA Managers’ Toolsets”, Vol. 8 No. 6, Journal of the FAA Managers Association.

[4] FAA ATC System Command Center Advisories database: http://www.fly.faa.gov/adv/advADB.jsp

[5] Anon., 2000, “Aircraft Situation Display To Industry: Functional Description and Interface Control Document,” Report no. ASDI-FD-001, Cambridge, MA, Volpe Center Automation Applications Division.

[6] Evans, Jim, Kathleen Carusone, Marilyn Wolfson, Bradley Crowe, Darin Meyer, and Diana Klingle-Wilson, 2001, "The Corridor Integrated Weather System (CIWS)," Portland, OR, 10th Conference on Aviation, Range, and Aerospace Meteorology.

[7] Exelis NextGen Data Subscription, http://www.exelisinc.com/solutions/nextgendata/Pages/default.aspx

[8] Anon., 2005, “Federal Meteorological Handbook No. 1: Surface Weather Observations and Reports,” FCM-H1-2005, Washington, D.C., NOAA.

[9] OPSNET Description: http://aspmhelp.faa.gov/index.php/OPSNET_Manual

[10] Anon., 2014, “FAA Order JO 7110.10 – Flight Services,” Chapter 9, Section 2: Pilot Weather Report (UA/UUA), FAA.

[11] Anon., 2014, “FAA ORDER JO 7110.10X10 – Flight Services,” Chapter 7, Section 1: AIREPs (POSITION REPORTS), FAA.

[12] Alexander, Curtis R., S. S. Weygandt, S. G. Benjamin, T. G. Smirnova, J. M. Brown, P. Hofmann, and E. P. James, 2011, “The High Resolution Rapid Refresh (HRRR): Recent and Future Enhancements, Time-Lagged Ensembling, and 2010 Forecast Evaluation Activities,” Seattle, WA, American Meteorological Society 91st Annual Meeting.

[13] TAF Description: http://en.wikipedia.org/wiki/Terminal_aerodrome_forecast

[14] TBFM Website: FAA TBFM Site: http://www.faa.gov/nextgen/snapshots/portfolios/?portfolioId=11

[15] Refer to http://www.pentaho.com/product/data-integration

[16] Lui, Max, 2014, “Complete Decoding and Reporting of Aviation Routine Weather Report (METAR),” TM-218385, NASA Technical Memorandum, NASA.

[17] Kimball, Ralph, Margy Ross, April 2002, “The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, 2nd Edition,“ Wiley Computer Publishing.

[18] Aviation System Performance Metrics 77 Airports, http://aspmhelp.faa.gov/index.php/ASPM_Airports

[19] Sridhar, Banavar, Sean S.M. Swei, 2007, “Classification and Computation of Aggregate Delay Using Center-based Weather Impacted Traffic Index,” AIAA 2007-7890, Belfast, Northern Ireland, 7th AIAA Aviation Technology, Integration and Operations Conference (ATIO).

[20] Refer to http://www.aviationweather.gov/adds/

[21] Reid, Michael, 2000, “An Evaluation of the High Level Architecture (HLA) as a Framework for NASA Modeling and Simulation,” 25th NASA Software Engineering Workshop.

[22] Swenson, H. N., T. Hoang, S. Engelland, D. Vincent, T. Sanders, B. Sanford, and K. Heere, 1997, "Design and Operational Evaluation of the Traffic Management Advisor at the Fort Worth Air Route

Traffic Control Center," Saclay, France, 1st USA/Europe Air Traffic Management R&D Seminar.

[23] Refer to http://www.w3.org/TR/html5/

[24] Refer to http://www.postgresql.org/

[25] Refer to http://postgis.net/

[26] Refer to http://www.opengeospatial.org/

[27] Refer to http://geoserver.org/

[28] DeLaura, R. and J. Evans, 2006, "An Exploratory Study of Modeling En route Pilot Convective Storm Flight Deviation Behavior," Paper P12.6, Atlanta, GA, American Meteorological Society's 12th Conf. on Aviation, Range, and Aerospace Meteorology.

[29] Refer to http://www.unidata.ucar.edu/software/thredds/current/tds/

[30] Refer to http://www.unidata.ucar.edu/software/thredds/current/netcdf-java/CDM/

[31] Refer to http://www.opendap.org/

[32] Grabbe, Shon R., B. Sridhar, and A. Mukherjee, 2013, "Similar Days in the NAS: an Airport Perspective," AIAA 2013-4222, Los Angeles, CA, 2013 Aviation Technology, Integration and Operations Conference.

[33] Mukherjee, Avijit, S. R. Grabbe, and B. Sridhar, 2013, "Classification of Days Using Weather Impacted Traffic in the National Airspace System," AIAA 2013-4403, Los Angeles, CA, 2013 Aviation Technology, Integration and Operations Conference.

[34] Grabbe, Shon R. and B. Sridhar, 2014, "Clustering Days with Similar Airport Weather Conditions," AIAA 2014-2712, Atlanta, GA, 14th AIAA Aviation Technology, Integration, and Operations Conference.

[35] Bloem, Michael and N. Bambos, 2014, "Ground Delay Program Analytics with Behavioral Cloning and Inverse Reinforcement Learning," AIAA 2014-2026, Atlanta, GA, 14th AIAA Aviation Technology, Integration, and Operations Conference.

[36] McNally, David, Kapil Sheth, et. al., 2013, "Operational Evaluation of Dynamic Weather Routes at American Airlines.", Chicago, Illinois, 10th USA/Europe ATM R&D Seminar

[37] Noy, Natalya, "Semantic Integration: a survey of ontology-based approaches," 2004, ACM SIGMOD Volume 33, Issue 4, ACM.

[38] Mayer-Schonberger , Viktor and K. Cukier, 2014, "Big Data: A Revolution that That Will Transform How We Live, Work, and Think", New York, NY, Houghton Mifflin Harcourt Publishing Company.

Acknowledgements

The authors would like to thank Mr. Pat O'Neal of UC Santa Cruz and Mr. Eric Wang of GTI Federal for their significant contributions to data acquisition, monitoring, and database administration.

Email Addresses

Michelle Eshow: Michelle.Eshow@nasa.gov

Max Lui: Max.Lui@nasa.gov

Shubha Ranjan: Shubha.Ranjan@nasa.gov