Appendix C

Engineering Design Document

Hazardous Weather Data Distribution Model

1 Table of Contents

2 Glossary of Terms, Acronyms, and Abbreviations

CIWS	Corridor Integrated Weather System: convective weather mosaics that provide forecasts up to 2 hours. Coverage in 2007: the highly congested Great Lakes and Northeast Corridors regions. Coverage in 2008: the continental U.S. and southern Canada
CWAM	Convective Weather Avoidance Model: statistical model that translates convective weather forecasts to flight deviation probability (DP) forecasts
DP	Flight Deviation Probability: a metric defined by CWAM as the percentage of aircraft expected to maneuver around regions of some specified weather conditions. It is inferred using statistical analysis and data mining techniques. Its units are in percent
DSS	Data Sharing Service
ETOP	Echo Top: an approximate measure of the cloud tops. This is used to infer information about nature of storm and potential severity. Echo tops are typically in the 5,000–70,000 ft range (with 55,000 ft being a more practical upper limit). Echo tops are typically geometric altitudes but are now translated to pressure altitude (in CIWS) for ATM use.
Deviation ProbabilityETOP	Flight Deviation Probability. See DPEcho Top: an approximate measure of the cloud tops. This is used to infer information about nature of storm and potential severity. Echo tops are typically in the 5,000–70,000 ft range (with 55,000 ft being a more practical upper limit). Echo tops are typically geometric altitudes but are now translated to pressure altitude (in CIWS) for ATM use
NEXRADDeviation Probability	Next-Generation Radar: a network of high-resolution doppler weather radars operated by the National Weather Service; also known as WSR-88D (Weather Surveillance Radar)Flight Deviation Probability. See DP
ThresholdNEXRAD	A term used to indicate a ‘cutoff level’ corresponding to a contour. It is used in a generic sense but for CWAM data it is the flight deviation probability. Next-Generation Radar: a network of high-resolution doppler weather radars operated by the National Weather Service; also known as WSR-88D (Weather Surveillance Radar)

VILThreshold	Vertically Integrated Liquid: a measure of the amount of liquid in a column of air obtained by integrating radar reflectivity measurements. A term used to indicate a 'cutoff level' corresponding to a contour. It is used in a generic sense but for CWAM data it is the flight deviation probability associated with a contour.
WATCARSVILThreshold	Weather And Traffic Complexity Avoidance Rerouting Software: A research study on NAS-wide mid-term strategic avoidance of weather with traffic complexity mitigation. Vertically Integrated Liquid: a measure of the amount of liquid in a column of air obtained by integrating radar reflectivity measurements. A term used to indicate a 'cutoff level' corresponding to a contour. It is used in a generic sense but for CWAM data it is the flight deviation probability associated with a contour
WATCARSVIL	Weather And Traffic Complexity Avoidance Rerouting Software: A research study on NAS-wide mid-term strategic avoidance of weather with traffic complexity mitigation. Vertically Integrated Liquid: a measure of the amount of liquid in a column of air obtained by integrating radar reflectivity measurements
WATCARS	Weather And Traffic Complexity Avoidance Rerouting Software: A research study on NAS-wide mid-term strategic avoidance of weather with traffic complexity mitigation.

3 Overview

This document describes the engineering design for basic weather facilities in ACES. A set of facilities will be provided to support the CWAM and support future extension to other weather products.

Figure 1 below depicts the role of the weather agent in ACES simulations. The weather agent is responsible for making weather data available to simulation models and updating it periodically. ATM models interact with the data via a set of interfaces that are typically contour oriented (i.e. data requests will return polygons). The weather agent dispatches a notification message (e.g. to WATCARS) after every update. The weather agent is also designed as a plugin for deployment flexibility.

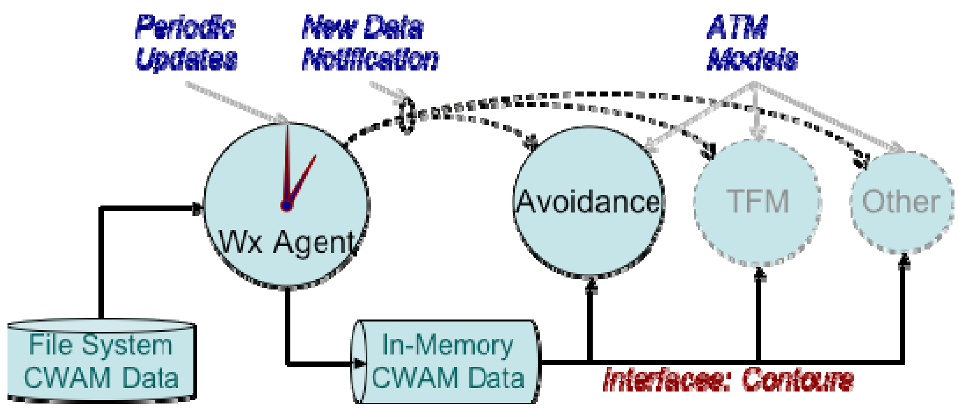

Figure 1. Role of the weather agent in ACES simulations

Basic weather facilities provide the following functionality:

1. **Data input:** loads data at specified intervals (unique to each weather product). Maps the time stamp for each loaded weather product to simulation time.
2. **Data model and query:** provides a flexible interface for representing and accessing data and makes data available to any agents that require its use.
3. **Reporting and Data Collection:** reports weather 'service' status and actions.

Note that real-time data feeds and visualization are not planned as part of this task. There are, in addition, several attributes in today's weather products that are of interest. These include

1. **Data latency**: Latency is defined as the delay between the time data are valid and the time they are available for use in client applications. This can influence model performance. For example, a five-minute delay can mean that the first forecast must be used instead of the nowcast.
2. **Dimensionality:** Weather data are multi-dimensional and often derive from radar data. The data include multiple forecasts spanning a known interval (2 hours in the case of CWAM).
   2.1. Some weather products are a function of altitude and are thus full 4D fields (e.g. CWAM), but some do not (i.e. they are 3D; e.g. CIWS, NEXRAD). In this task we are only concerned with CWAM.
3. **Format:** Weather data is typically gridded or contoured. Grid data are defined at points on a rectangular grid (in a given map projection). Contour data are regions (polygons) with values at or above a specified threshold.
4. **Airspace and/or Region:** Weather data may be restricted to specific airspace or locality (e.g. CWAM is an en route airspace model currently forecasting the eastern United States—soon to forecast the entire Continental US).
5. **Uncertainty:** Weather phenomena are stochastic by nature. Uncertainty will not be addressed in this task.

This task will provide the functionality required to support simulations using CWAM weather data as described in the remainder of the document. WATCARS, which is a weather avoidance model for ACES, is one of the models that will use CWAM; it is currently in development.

3.1 CWAM

Of the various weather products available, CWAM has been proposed for use in managing weather impacts in ATM in the 0 to 2 hr strategic domain. CWAM is a parametric model (from MIT Lincoln Labs) that correlates convective weather phenomena with likelihood of aircraft deviation in en route airspace. Data mining and pattern classification of weather and traffic data are used to generate the model parameters. The model output is a probability of deviation at each (3D) location in the airspace. This output—a four dimensional field—is compressed into a set of contour maps (at various flight deviation probability thresholds) that span the 25,000–45,000 ft altitude range (typical) and the 0–2 hour forecast range. Altitude resolution is 1,000 ft and forecast resolution is 5 minutes giving a total of 21 altitudes, one nowcast, and 24 forecasts. CWAM data are updated every 5 minutes.

CWAM data are stored as HDF5 files that are currently generated offline (i.e. not in real time). A brief description of HDF5 is given in the 'HDF5 Library' appendix. The format of CWAM files is summarized in the 'CWAM Data Format' appendix.

CWAM uses CIWS data (available in real time from MIT Lincoln Labs) to generate the deviation probabilities. CIWS provides forecasts for Vertically Integrated Liquid (VIL: 1–6) and cloud echo tops (ETOP) up to 65,000 ft. The appendix 'CIWS and CWAM Sample Images' provides CIWS and CWAM sample images. Note that while CIWS data are inherently two dimensional (spatially), CWAM data are three-dimensional. Furthermore, CIWS data are meteorological whereas CWAM data are ATM specific. For details on the CWAM model, see "Referenced Documents 1 & 2" below.

3.2 Referenced Documents

1 DeLaura, R., and Evans, J., "An Exploratory Study of Modeling Enroute Pilot Convective Storm Flight Deviation Behavior," Proceedings of the 12th Conference on Aviation, Range, and Aerospace Meteorology, Atlanta, 2006.

2 Chan, W., Refai, M., and DeLaura, R., "An Approach to Verify a Model for Translating Convective Weather Information to Air Traffic Management Impact," 7th AIAA Aviation Technology, Integration and Operations Conference (ATIO), Belfast, Northern Ireland, 18 - 20 September 2007.

4 Requirements

Table 1. General CWAM Requirements (primary requirements are highlighted)

ID	Requirement
1	The system shall allow the user to specify a directory for loading CWAM data files (via scenario files or configuration)
2	CWAM data shall be stored on a shared file system.
3	CWAM data for a test or experiment shall be provided in advance.

4	**The system shall load CWAM data files at known intervals (5 minutes of simulation time). The time stamps of the loaded data shall be mapped to simulation time.**
5	The system shall provide a mechanism for versioned data loading. The system shall check the data version and use the appropriate reader for the version. One reader shall be implemented to support the current format.
6	**The system shall dispatch a message upon CWAM data update.**
7	**The system shall dispatch a failed message if data loading fails.**
8	The system shall maintain the full forecast data (2 hours worth; i.e. 1 nowcast and 24 forecasts) for each loaded file.
9	**The system shall provide support for contoured multi-dimensional data. See §5.2 for details; also see requirement 12 below.**
10	Weather contour representation shall provide a valid time stamp, a forecast time, an altitude, and a threshold for each loaded polygon.
11	Polygons shall provide bounding boxes. These shall be calculated on load and are used to facilitate quick culling of polygon–trajectory intersections. An interface for checking intersections or containment within the bounding boxes shall be provided.
12	**Weather classes shall provide an interface that supports queries on forecast, altitude, threshold or combinations of those. Specifically, and at a minimum, a query on all three shall be provided. These queries return polygon data to the requestor.**
13	Each container shall use shared weather data to minimize network traffic and memory footprint. Each container shall maintain its own copy if and only if a weather agent is instantiated in the container.
14	Each weather product type shall be identified and accessed using a key.
15	Design shall comply with ACES architecture paradigms. and the plugin mechanism shall be leveraged.

Table 2. Stretch goals

S1	*Stretch goal:* The system can support loading a subset of the available data. For example, it may be desirable to load a single threshold used in detection and resolution. The benefit is reducing the memory footprint.
S2	*Stretch goal:* A script can be provided to copy weather files from repository (shared file system) to each node’s hard drive.
S3	*Stretch goal:* The system can allow the user to specify the latency (fixed or—optionally—random sequence) for shifting the CWAM data time relative to simulation time (to simulate real-time data latency).

5 System Design

The general approach is to provide a weather agent and a data processing activity[1] that performs the various processing steps required to load data and share it with other agents. The weather agent will be distributed to simulation containers that instantiate models that use CWAM data.

5.1 Weather Agent

The weather agent is the agent within which the input processing activity (LoadWxActivity) runs. It is responsible for sharing the loaded data with other agents or activities and for handling failed loads (in this task failed loads will simply be ignored). The agent handles a ‘DataLoaded’ message (during which it will share the new weather object).

In general, the steps used to process weather data are:

1. Check availability—Determine if new data is available from the source.
2. Locate—Identify the location of or means for obtaining the new data. Often, but not always, this can be combined with step 1.
3. Load—Parse new data and load into object.
4. Sync—Apply time synchronization policy to the new data; in other words map data time to simulation time. (e.g. apply latency).
5. Share publicly—Share the object and notify other agents and activities that new data is ready for use. Once the object is shared publicly no further modifications are made to it.

Figure 2 below depicts these steps as a sequence diagram (excluding optional actions).

[1] In the future, when multiple weather products are desired, there will be multiple such activities, one per weather product.

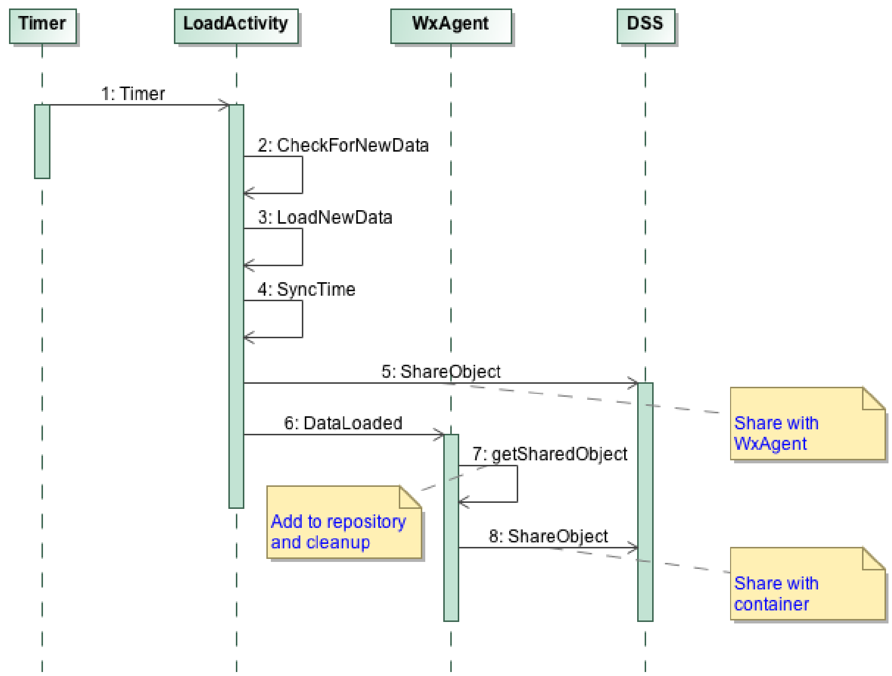

Figure 2. Data Processing

5.1.1 LoadWxActivity

The 'LoadWxActivity' is a child of WxAgent and is responsible for testing for availability of new CWAM weather data, loading the new data when available, and for mapping the data time to simulation time. This activity supports pre-processed data made available on a file system.

The activity handles a discrete timer event (every 5 minutes by default). For each timer event the activity will determine if new data is ready to load. It identifies the appropriate data file to read based on the simulation time and the file time stamp. The data are loaded into a weather object then the data time is mapped to simulation time. The activity will share the object loaded with the WxAgent and dispatch a 'DataLoaded' message upon successful data load.

This activity addresses requirements 4, 6, 7, and 13 and depends on requirements 1, 2, 3, and stretch goal S3 if implemented.

5.1.2 Mapping Data Time to Simulation Time

Configuration settings for the activity include a data directory, a start time, the first CWAM time stamp to load, and *optionally* latency. This information is used to map simulation time to CWAM time (i.e. determine what files to load and how to time stamp the loaded data).

If T_S is the start time, T_{CWAM} is the first CWAM time stamp to load, and ΔT_L is the latency, then at the n^{th} load cycle, CWAM data at time stamp ($T_{CWAM} + n \times 5\ minutes$) will be loaded. The CWAM data is given a simulation time stamp of ($T_S + n \times 5\ minutes - \Delta T_L$).

5.2 Data Representation

Generally speaking, weather data are 4 dimensional fields that are updated periodically and can have many data representations. Today's weather data are typically grid based. Radar based data, for example, are essentially a group of images each representing a 2 dimensional view of a weather phenomenon. Regardless of internal representation, however, it is beneficial to provide 'slice views' into the data. A slice view is one that returns a subset of the data given some 'slicing' parameters. The simplest slicing policy is to slice on forecast, time, and altitude. This is also consistent with radar-based data sets. An additional slicing strategy uses a 'threshold' to provide a contour of a desired phenomenon. In this task we will support contoured multi-dimensional data and will provide a contour-based slice view for loaded CWAM data (requirement 9).

The weather representation will provide an interface for requesting data views. Data can only be queried though a view returned by the interface. These views will provide an interface for requesting data given forecast, altitude, and threshold (requirements 10 and 12). In general, a weather field that does not support a requested view can return null views (or, *optionally*, dynamic views that are generated on the fly). Dynamic views are not supported in this task. Note that the views supported by a weather product are specific to that weather product. Some might provide contours, others raw pixel maps, and so on.

5.3 Loading CWAM Data From File and Versioning

CWAM data are packaged in HDF5 binary files. This format is subject to change so this task will provide a mechanism to support versioned data loading (requirement 14). The task, however, will only implement the current format; other formats can be added as they become available.

To load a file, a generic CWAM file reader will read the version information. The version information will be used to request the appropriate reader, which is then used to process the data. In the absence of version information the generic reader will process the data based on the format identified in the 'CWAM Data Format' appendix. Figure 3 below depicts a simplified class diagram for the reader hierarchy.

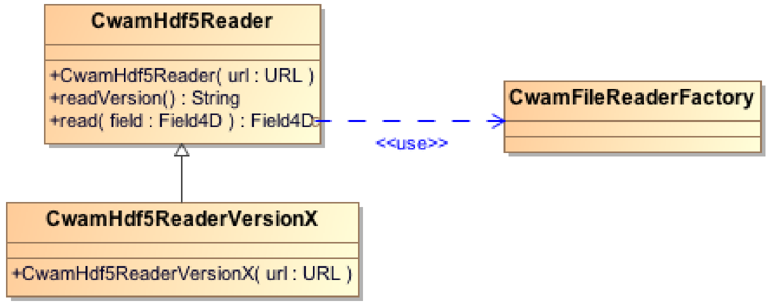

Figure 3. Versioned Readers

5.4 Polygon Properties

Requirement 11 above identifies the bounding rectangle property used to assist in culling polygon and trajectory encounters. There are various other properties that might be of use in detection and resolution algorithms (including centroid, area, convex hull, and so on) but we do not address those here. Note that in the remainder of this section we use plane geometry (e.g. map projection).

In general a polygon can be bound by rectangles at various orientations (the minimum bounding rectangle being particularly desirable when culling polygons but more expensive to compute). Two orientations are commonly used for collision detection, namely upright rectangles and 'diamonds'. Figure 4 below depicts how the bounds are used to determine if a polygon might contain a point by comparing to a bounding rectangle at a given orientation. This task will provide upright ($A = 0,\ B = 1$) and optionally 45° ($A = 1,\ B = 1$) bounding rectangles (the latter can be used for additional culling). Containment is inclusive of boundary edges.

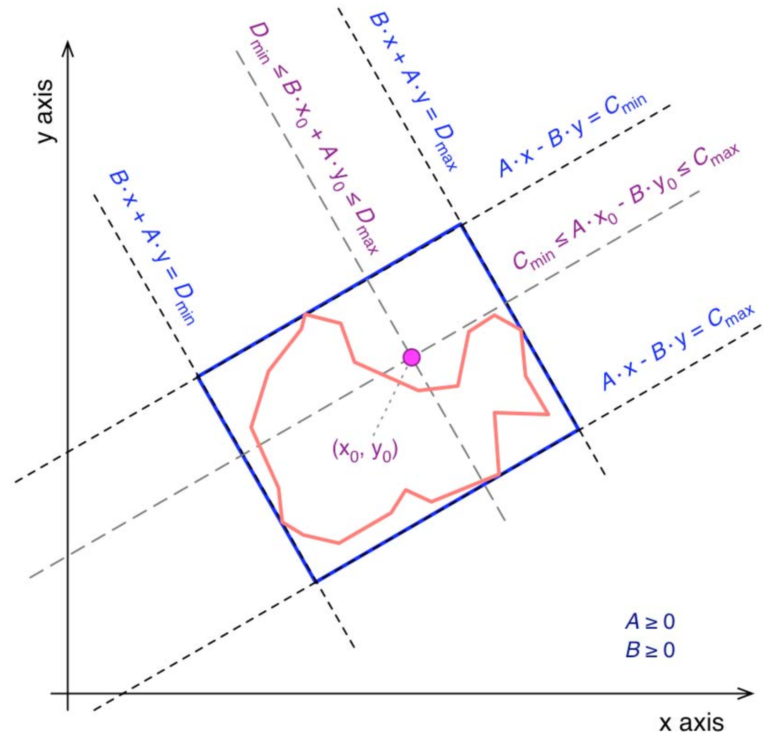

Figure 4. Containment within polygon bounds

5.5 Data Sharing

The Weather Agent will maintain a repository of weather data and will share the data using the Singleton pattern. Note that weather data sizes can be significant (on the order of 100–200 MB or more) for heavy weather.

5.6 CWAM Configuration

Configuration parameters include the file system directory (where data files are stored), start and end times (in simulation time), CWAM start time (valid time—UTC). They can optionally include latency and selected forecast, altitude, and threshold ranges to load.

5.7 Failure Modes

CWAM data failures are reported but no special handling or error recovery is planned. CWAM data may fail to load under the following conditions,

- Data files are ill-formed or corrupted
- Data file format is not supported
- Data files are missing.

6 Stretch Goals

6.1 Simulation Data

A script can be provided to copy data to local file systems for each node in a simulation. The motivation behind this is that CWAM data files can be large in size (100MB for heavy convective weather) and the use of a shared file system can be a burden on the network.

6.2 Data Latency

Data latency[2] affects the detection and resolution agents. In general real time weather data takes time to process. The latency resulting from CWAM processing for example is on the order of 5-10 minutes. This means that the nowcast is already old by the time the data is available for use in detection and resolution. To illustrate the role of latency in a simulation assume that the data has a 5-minute delay. This means that detection and resolution models cannot rely on the nowcast for evaluating current weather conditions. The 5 minute forecast might be more appropriate; a similar offset would also apply to all forecasts not just the nowcast. One may argue that fast time simulations do not need to worry about data latency because the real data is already available; this, while true, does not address the need to evaluate model performance in the presence of latency. For this reason it would be of interest to provide support for simulating latency in ACES experiments.

Supporting latency is simply a matter of providing configuration to set latency policy. Latency policy includes a nominal latency, at a minimum, and can also include a random deviate to allow for jitter. See §5.1.1 (LoadWxActivity), above for details on how latency affects the data time stamp.

Latency for real-time data is a characteristic of the data and no user configuration is required for setting it. For real-time simulations the data time stamp remains unchanged.

7 Limitations and Challenges

This work does not address generation of CWAM data nor does it address real-time simulation concerns. CWAM data are generated offline using an existing utility. The basic challenge is in sharing and synchronizing the large and periodically changing CWAM data sets among several nodes within the simulation. Sharing this data via message passing is not practical, which is why sharing is done through a Data Sharing Service. Using a centralized database was considered, but the high latency and overhead of each simulation node accessing large volumes of data from the central database would unacceptably impact simulation performance.

8 Data Collection

The objective of data collection is to provide diagnostic information on the data loaded throughout the simulation. Of interest are the data file names, the times they are loaded,

[2] Latency is also known as time-delay in systems control terminology.

the status of each load operation, the simulation time stamp assigned to each loaded file and so on. Additional information not planned for this task includes profiling data.

1 Collect CWAM configuration data (in a new table cwam_config).
2 Collect data on loaded files, including the file name, timestamp, and status (success or failure). Also record the simulation time stamp assigned to the loaded data. This information will be stored in a new table, cwam_load_info, in the database.
3 Log any failures when loading CWAM data files, including the file name, timestamp, and the error message.
4 Optional: data message that includes polygon vertices in latitude and longitude along with altitude bounds, time-stamp, and level of hazard. Note that collecting all polygons might be prohibitive in terms of data size so only select polygons would be collected.

9 Test Approach

Weather data for July 27th, 2006 will be made available and used for testing the weather facilities provided. This date is a heavy weather day with interesting weather patterns.

The objectives include testing for the preservation of system invariants (such as update interval, system response to successful or failed loads, messages sent and received, and so on). Also of interest is coherency across multiple containers and performance profiling (such as time to load data, impact on memory footprint and garbage collection, and so on).

The following types of tests will be run to validate the requirements:

9.1 Loading CWAM Data

1 Verify that the system loads CWAM data files from the directory specified in he scenario file or configuration. Use alternate directories. [EDD Requirement 1, 2]
2 Verify that the system handles missing data files properly. [EDD Requirement 3]
3 Verify that the system loads the appropriate CWAM files at known intervals (5 minutes by default). Confirm that the system maintains the full set of forecast data. [EDD Requirement 4, 8]
4 Verify that the system reports and logs an error message if CWAM file load operation fails. [EDD Requirement 7]
5 Verify that the system can query the CWAM data. [EDD Requirement 10, 12]
6 Verify that the system identifies the appropriate data to load at a specified simulation time. [EDD Requirement 4]
7 Verify that the system supports versioned data loading. This is to be accomplished via logging (since only one reader will be implemented). [EDD Requirement 5]

9.2 Sending Messages to Other Agent

1 Verify that the system sends a message after the CWAM data is loaded. [EDD Requirement 6]

9.3 Unit Tests for Weather Data APIs

1 Test that the APIs provide a valid time stamp, a forecast time, an altitude, and a threshold for each loaded polygon. The last three define the degrees of freedom for the contoured weather data.
2 Test that the all vertices in a polygon are within its bounding boxes.
3 Test that the APIs support queries on forecast, altitude, threshold or combinations of those.
4 Test that CWAM weather interfaces are accessed using a key.

9.4 Other Tests

1 Verify that each container maintains its own copy of weather data (given a weather agent in said container). Verify that data is not duplicated within sharing scope.
2 Verify that ACES deployment includes all the required HDF5 library files for the correct architecture. Verify that the environment is setup properly for the architecture.
3 Review the design and confirm that it complies with ACES architectural paradigms.
4 *Stretch goal:* Test support for partial data loading.
5 *Stretch goal:* Test the script that copies weather files from repository to each node's hard drive.
6 *Stretch goal:* Test that the system loads files with the right time stamp when using latency.

9.5 Stress Test

Run ACES with a full data set and load CWAM data over a period no less than 24 hours. *Optional:* Profile the memory and performance of the load operation and note its dependence on the amount of data loaded and simulation time.

1 Test that the overall system performance is not increasingly degraded as the system loads CWAM data repeatedly.

9.6 Multi-Platform Test

Repeat the tests of §9.5 for all supported platforms.

10 Appendices

10.1 CIWS and CWAM Sample Images

The following figures are provided for illustration purposes. Section 3.1 above describes these figures and the relationship between CIWS and CWAM.

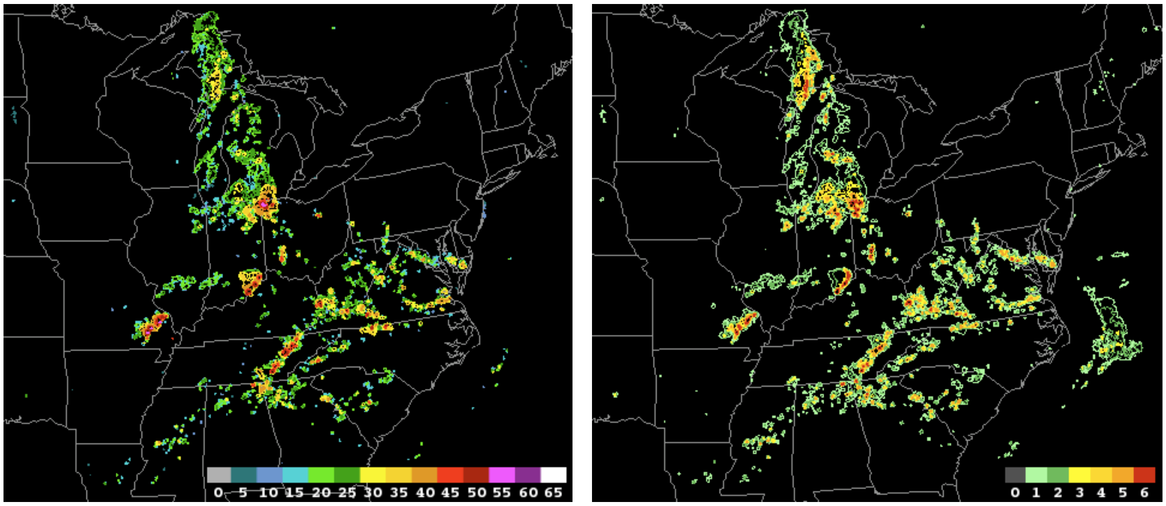

Figure 5. CIWS Echo Tops and VIL

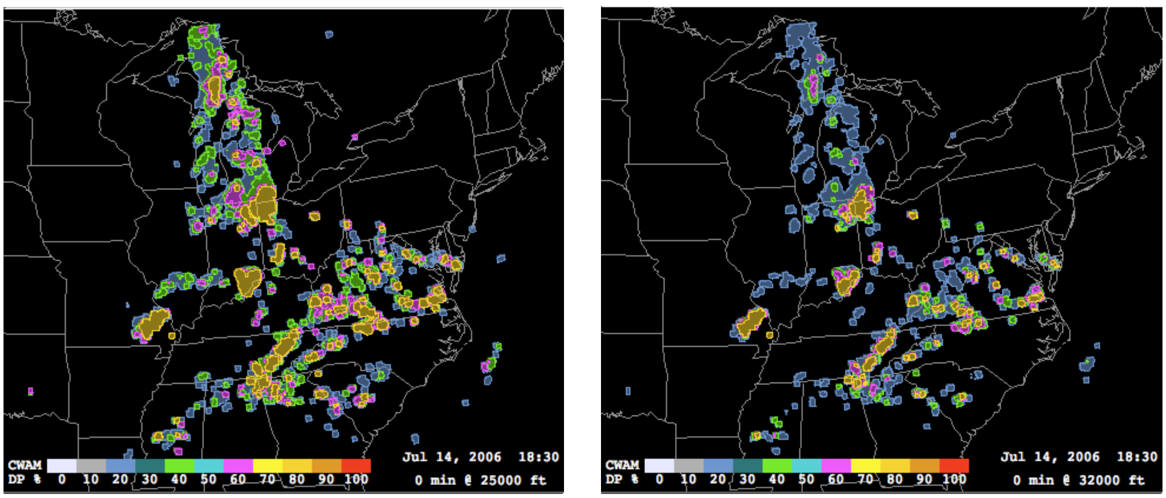

Figure 6. CWAM Deviation Probability at 25,000 ft and 32,000 ft respectively

10.2 HDF5 Library

The HDF5 library supports a hierarchical data format for file system based storage of large-scale data. The library (Fortran, C, C++, and Java interfaces) is available from http://www.hdfgroup.org/.

HDF5 provides a specification for a binary file format that is platform independent and supports query and object discovery through interfaces that are provided by the library. It is available for various platforms including MS Windows, Mac OS X, Linux, and others. The library provides several useful utilities two of which are of particular interest; namely, HDFView (a graphics tool for browsing and viewing HDF5 data) and h5dump (a command line utility that can be used to collect format information or dump file contents in readable form). These utilities are not required for supporting CWAM in ACES but are useful because they allow the user to inspect the contents of data files in a readable format. Please see the above link for additional information.

10.3 CWAM Data Format

CWAM data are provided as contours of flight deviation probability at various forecasts, altitudes, and thresholds (% deviation probability). Data are stored in HDF5 files and are formatted hierarchically as described in Table 3 below. File names include a time stamp that is identical to the 'valid time' for the data. The file names are typically formatted as follows: 'yyyy_mm_dd_HH_MM_GMT.CSFD.h5,' where y stands for year, m for month, d for day of month, H for hours, and M for minute.

Table 3. CWAM file hierarchy[3] (sample file '2006_07_14_17_05_GMT.CSFD.h5')

HDF5 Object4	HDF5 Name5	Description6	Value Type
GROUP	"/"	Root group	–
GROUP	"Deviation Probability"	Contains Contour & optional Raw Data	–
ATTRIBUTE	"Valid Time"	Forecast GMT time	String
GROUP	"FCST###"	### ≡ forecast minutes [000:5:120] Contains Flight level data	–
GROUP	"FLVL###"	### ≡ flight level [250:10:450] Contains Contour & Raw Data	–
GROUP	"Contour"	Contains Threshold data	–
GROUP	"TRSH###" …	### ≡ threshold [010:010:100] Contains Polygon data	–
ATTRIBUTE	"Threshold"	Threshold value [10:10:100]	Integer
DATASET	"POLY####" … Vertex latitudes7 Vertex longitudes	#### ≡ polygon number [0000:1:*] Polygon data contains vertex latitudes and longitudes	Float array

[3] Some nodes in the file hierarchy are omitted for simplicity.
[4] Indentation indicates hierarchical relationship to object above it
[5] # stands for digit; e.g. "FCST###" can be "FCST000", "FCST005", …, "FCST120"
[6] The notation [*a*:*b*:*c*] indicates range from *a* to *c* in *b* increments (inclusive)
[7] Latitudes and longitudes are given in degrees. Longitudes adopt the East +ve / West –ve sign convention.