A Data Manager to Multicast UAS IDS Data to Multiple IDSs

Ron Marsh, Ph.D., Kirk Ogaard, and Micah Kary
Computer Science Department,
University of North Dakota
Grand Forks ND 58203
rmarsh@cs.und.edu
kirk.ogaard@und.nodak.edu
micah.kary@und.nodak.edu

John Nordlie
Regional Weather Information Center,
University of North Dakota
Grand Forks ND 58203
nordlie@rwic.und.edu

Abstract

The University of North Dakota, in cooperation with the Federal Aviation Administration, is developing airspace within the state of North Dakota where unmanned air systems can be tested/operated. The John D. Odegard School of Aerospace Sciences, with funding from the United States Air Force, is developing a ground-based radar system capable of detecting aircraft while developing the software to optimally display the information to aircraft operators.

The system will integrate aircraft position data from Automatic Dependent Surveillance-Broadcast, ground based radar and telemetry data from global positioning system equipped aircraft and will display the data on display systems. A Data Manager will poll the sensors collecting the data, sources for weather information relevant to the monitored airspace and multicasts the data to a Range Control Safety and Ground Observer Information Display Systems.

This paper provides an overview of the Data Manager, the simulation of the operational airspace, and the display systems.

1. Introduction

The University of North Dakota (UND), in cooperation with the Federal Aviation Administration (FAA), is identifying airspace within the state of North Dakota where organizations interested in developing unmanned air systems (UASs) can test/operate their systems without the need for an on-board sense and avoid system. Taking advantage of a relatively low population density, the UND and the state of North Dakota are working to provide more than 13,000 square miles of airspace suitable for all manner of UAS operations without the need for implementation of temporary flight restrictions (TFRs) [1].

The John D. Odegard School of Aerospace Sciences, with funding from the United States Air Force (contract number FA4861-06-C-C006), is developing a ground-based radar system capable of detecting low observable aircraft such as sailplanes and hot-air balloons while developing the software to optimally display the information to operators of UAS. Where previously available ground-based radar systems have not been sufficient for the FAA to approve their use for sense and avoid mitigation, this system will employ new technology that will enable UAS operators to see potential conflicts before they become a problem and safely maneuver their craft away from non-cooperative aircraft. Sophisticated algorithms are being developed to determine optimum scan patterns, rates and data assimilation to provide the most comprehensive "picture" of the operating environment.

Funding of the program is allowing the UND to develop a comprehensive system that combines a plethora of data into a “big” picture. Ultimately, the UND expects to provide an interim mitigation strategy to allow UAS research and development outside restricted airspace thus aiding the FAA in its efforts to develop appropriate regulations relating to UAS operations and certification.

2. Background

The rapid development of UAS in the United States and worldwide has generated investment in research, technologies and systems at an unprecedented rate. The proven successes of unmanned aircraft (UA) in the military environment and rapid advances in commercial navigation, automation and sensing technologies have created new opportunities for civilian applications of UAS. These opportunities have in turn imposed ever-increasing pressure on the FAA to respond with certification standards and regulations that will allow UAS routine access to the national airspace (NAS) [2]. These standards and regulations, by necessity, will apply to, not only the public and civil operators; they will apply to military UAS as well.

Overshadowing all areas of regulatory effort is the FAA’s mandate to quantify and achieve a target level of safety that is consistent with the current level of safety of manned aircraft with respect to ground fatalities and mid-air collisions. We must assume that a ground impact may result in fatalities, and is therefore categorized by FAA criteria

[3] as a hazardous event. The target level of safety of a hazardous event, as specified by FAA requirements for manned aircraft is 10^{-7} events per hour of operation [4, 5, 6]. The target level of safety for the midair collision hazard from FAA safety guidelines for manned aircraft is 10^{-9} collisions per hour [3]. The obstacle to progress, however, is the apparent inability to "safely" integrate UAS into the NAS. "An equivalent level of safety" is a term heard often today. It implies that, in part, any system utilized in the airborne vehicle that effectively replaces a particular capability of the pilots left on the ground, their ability to "see and avoid" other aircraft, must be as good or better than the pilot it replaces [7].

Policy Memo 05-01 [7] gives military, public and private operators of UAS guidance on how operations are to take place and what safety mitigations are allowed to be employed for Sense and Avoid (SAA). Without an onboard SAA system, the operator may use observers on the ground provided the UA is flown below three thousand feet above the ground (AGL) and within one mile laterally. If the UA is to be flown higher or farther from the observer, the UA must be chased by aircraft with an on-board observer. If the UA is flown in restricted airspace, however, no specific means of SAA is needed due to the nature of the airspace.

Although most UAs will be equipped with global positioning system-based technologies, such as Automatic Dependent Surveillance – Broadcast (ADS-B), or other standard navigational aides, such technology cannot be assumed to be in place on other light aircraft operating in the same airspace as the UA. This argues for the need for additional non-line-of-sight methods for monitoring the complete airspace desired for UAS deployment. Operationally, radar systems have been used for the detection of aircraft for over fifty years. Recent advances in radar technology, along with corresponding advances in communications, computing, and data processing, now provide the capability for real-time surveillance of large volumes of the atmosphere.

3. UND Risk Mitigation Architecture

The airspace risk mitigation system is meant to be an extension of the ground-based observer. The system will integrate aircraft position (latitude, longitude, and altitude) data from sources such as Automatic Dependent Surveillance-Broadcast (ADS-B) [8], ground based radar, and telemetry data from Global Positioning System (GPS) equipped aircraft. Aircraft position data will be fused and forwarded to the Range Control Center (RCC). Data from a weather station located at the UAS operations airport and Doppler weather radar would also be forwarded to the RCC. At the RCC, a Data Manager (DM) acquires the variety of data and multicasts the data to the display systems, including a high resolution Range Control Center Information Display System (RCC IDS) and one, or more, high resolution Ground Observer Information Display Systems (GO IDS). The RCC IDS, which is modeled after existing Air Traffic Control (ATC) display systems and existing Traffic Information Service-Broadcast (TIS-B) [9] display systems**,** displays the georeferenced GPS positions of all aircraft operating in the area, the georeferenced positions of ground-based hazards/targets, weather information, system health data, and the risk parameter. The GO IDS, which is modeled after existing Flight Information Service-Broadcast (FIS-B) [9] moving map display systems, portrays the positions of all aircraft operating in the area in relation to a specific aircraft of interest, weather information, system health data, and the risk parameter. The prototype architecture for the UND risk mitigation system is shown in figure 1.

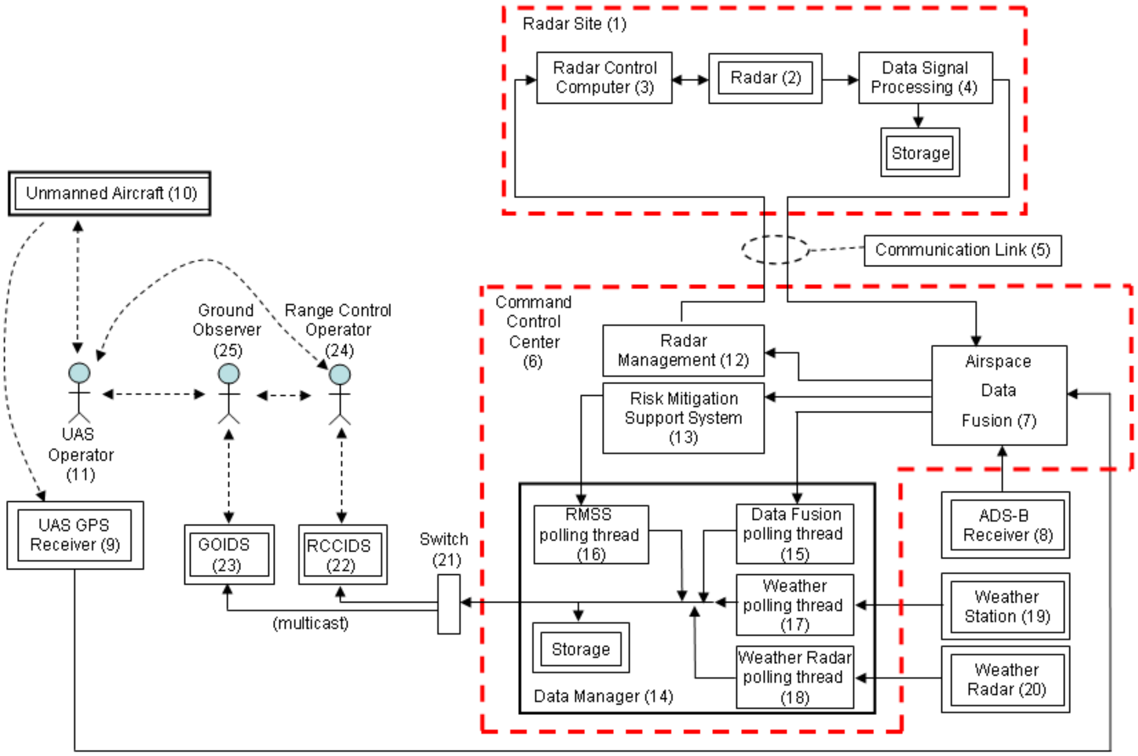

Figure 1: UND risk mitigation prototype architecture.

The goals of the information display systems (IDSs) are to provide Range Control Center (RCC) personnel and Ground Observers (GOs) with pertinent and timely information such that the safe operation of UAs is possible. Since we currently do not have the applicable sensors/receivers (radars or ADS-B receivers), our development work has been limited to the development of the display systems and simulation of the airspace.

As noted above we are limited to simulating the airspace and we have attempted to do so in a manner most like that of the envisioned system. In the actual system, it is expected that a data fusion system will combine the radar and ADS-B data. Thus, we have simulated that data stream on a separate computer that streams the information to our display system over a network. In the actual system, it is also expected that we will receive UAS position information from the GPS equipped UA via a telemetry stream. Thus, we have simulated that data stream on another separate computer (running the open source flight simulation software “FlightGear”) that streams the information to our display system over a network. We are also acquiring near real-time weather information from a website hosted by the Regional Weather Information Center (RWIC) and Doppler weather radar data from the National Weather Service (NWS). It should be noted that we originally used Microsoft's Flight Simulator X (FSX) for all of the airspace simulations, but ran into insurmountable difficulties in getting FSX to communicate with the Linux-based display systems. The simulation facility is shown in figure 2.

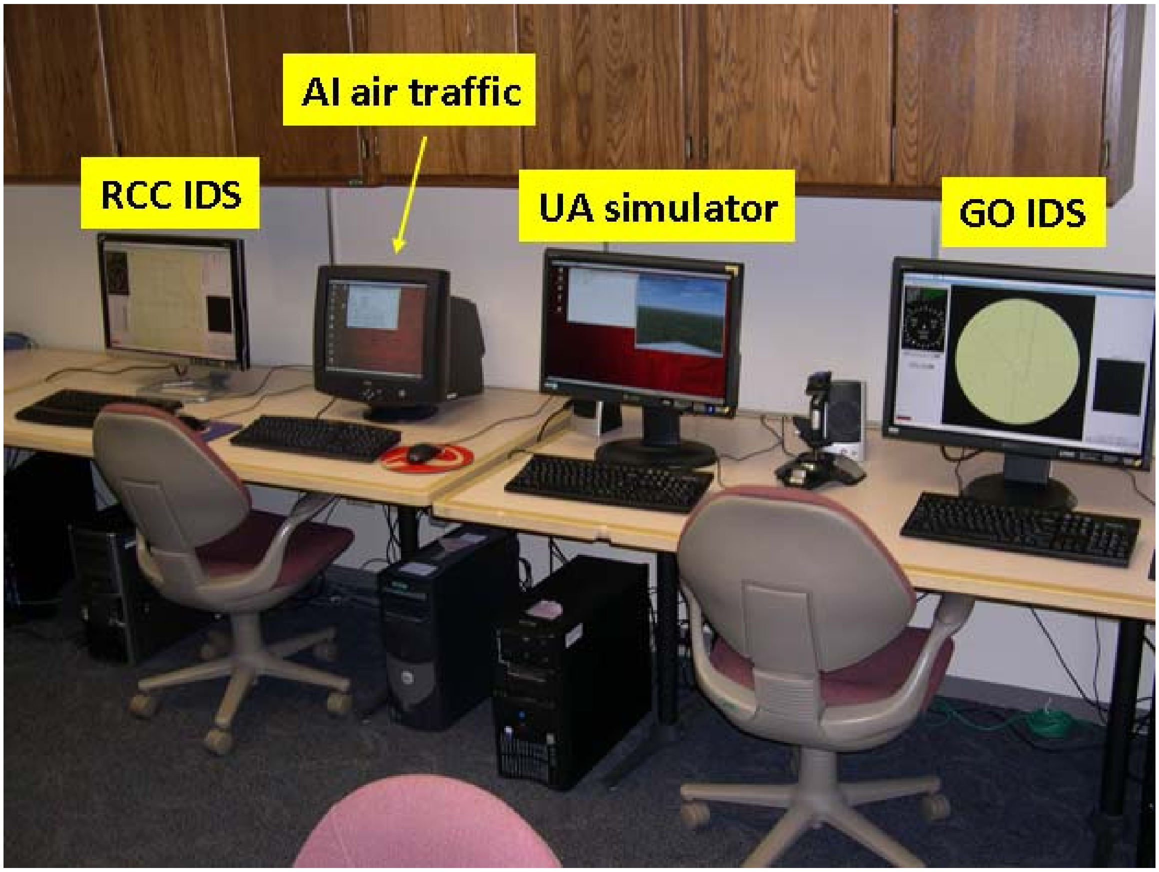

Figure 2: UND risk mitigation simulation lab.

4. Data Manager

The DM runs 5 threads: 4 data acquisition and packaging threads and 1 multicasting thread. There are separate threads for acquisition and packaging weather data (e.g. temperature, humidity, and wind speed), the Doppler radar image, data from the AI traffic generator, and data from the FlightGear/UAV simulation. The weather data thread downloads data (via GNU's Wget) from a weather station database approximately every 2 minutes. The Doppler thread downloads (via Wget) a Doppler weather radar image from the National Weather Service's website approximately every 7 minutes. The AI thread listens for connection requests from the AI traffic generator. Once it has a connection to the AI traffic generator, the AI thread starts receiving and packaging the information about the aircraft currently being simulated by the AI traffic generator. The AI traffic generator provides updates approximately every second. If the connection to the AI traffic generator is lost, the AI thread waits for the next connection request. The UAV thread is similar to the AI thread, except it connects to the FlightGear/UAV simulation. As data is received, it is sent to the IDSs twice per second using a multicast approach via the multicasting thread.

The problem with developing a system like this is that it *must* broadcast accurate and complete data, it *should* broadcast on a timely manner and the system *must* be robust. Given that our weather data sources are currently only accessible via Wget, and that Wget “fails” often (due to database updates at the source site), both the weather and Doppler acquisition threads use the UNIX system command “System” to execute stand alone programs to delete old local versions of the data files and to retrieve (via Wget) the latest required data files. If a stand alone program successfully retrieves the data, there will be a local data file(s) for the applicable thread to open. If a stand alone program fails (due to Wget failing, etc), there will not be any local data file(s) for the applicable thread to open. Thus, the design provides the two threads with a simple mechanism for determining if a new data file has been retrieved via Wget.

The DM communicates with the AI traffic generator and FlightGear/UAV simulation using BSD-style sockets. The AI and UAV threads perform the same socket operations on different ports. Each thread uses two sockets, since the DM doesn't permit multiple connections to its ports at the same time, two ports (one per socket) are used. Both sockets use the TCP/IP connection-based protocol from the IPv4 protocol family. The thread creates the first socket to monitor the appropriate port for a connection request from the client (i.e. the AI traffic generator or the FlightGear/UAV simulation). Then the thread waits until it receives a connection request. Once the thread receives a connection request, it accepts the request and creates the second socket. The second socket is used to receive data from the client. Once the connection to the client is established, the first socket is no longer needed.

The AI thread, for example, creates the first socket by calling socket(), binds it to the appropriate port (e.g. port 5501) by calling bind(), and puts it into the listening state by calling listen(). Then the AI thread calls accept(), which waits until it receives a connection request from the AI traffic generator on the port and then creates the second socket. When the AI thread is ready to receive data from the AI traffic generator, it calls recv() with the second socket as a parameter. While the thread remains connected to the AI traffic generator, the thread continuously polls the socket for new data items from the generator. The data items are received from the socket as raw sequences of characters terminated by newlines. Each data item is then parsed from the raw sequence of characters into its internal form, such as a floating point number. If the thread loses its connection to the generator, it will wait indefinitely until the connection to the generator is reestablished.

The data packaging threads collect data from their respective sources, and then store the data in local data structures. Once a complete set of data has been received by a data packaging thread, it copies the contents of its local data structure into the shared data structure used by the multicasting thread. Since the DM runs 5 threads concurrently, semaphores are used to insure that the multicasting thread doesn’t multicast incomplete data over the LAN. Each of the 5 threads has an associated semaphore. When one of the data packaging threads is ready to copy its local data into the shared data structure, it waits for the multicasting semaphore to clear. Once the multicasting semaphore has cleared, the data packaging thread sets its semaphore to indicate that it's copying into part of the shared data structure and the multicasting thread shouldn‘t multicast it. After the data packaging thread has finished copying its local data into the shared data structure, it clears its semaphore. The multicasting thread waits for all semaphores associated with data packaging threads to clear. Once the semaphores have all cleared, the multicasting thread sets its semaphore to indicate that it is multicasting the contents of the shared data structure and none of the other threads should modify it. After the multicasting thread has multicast the contents of the shared data structure over the LAN, it clears its semaphore. The DM architecture is shown in figure 3.

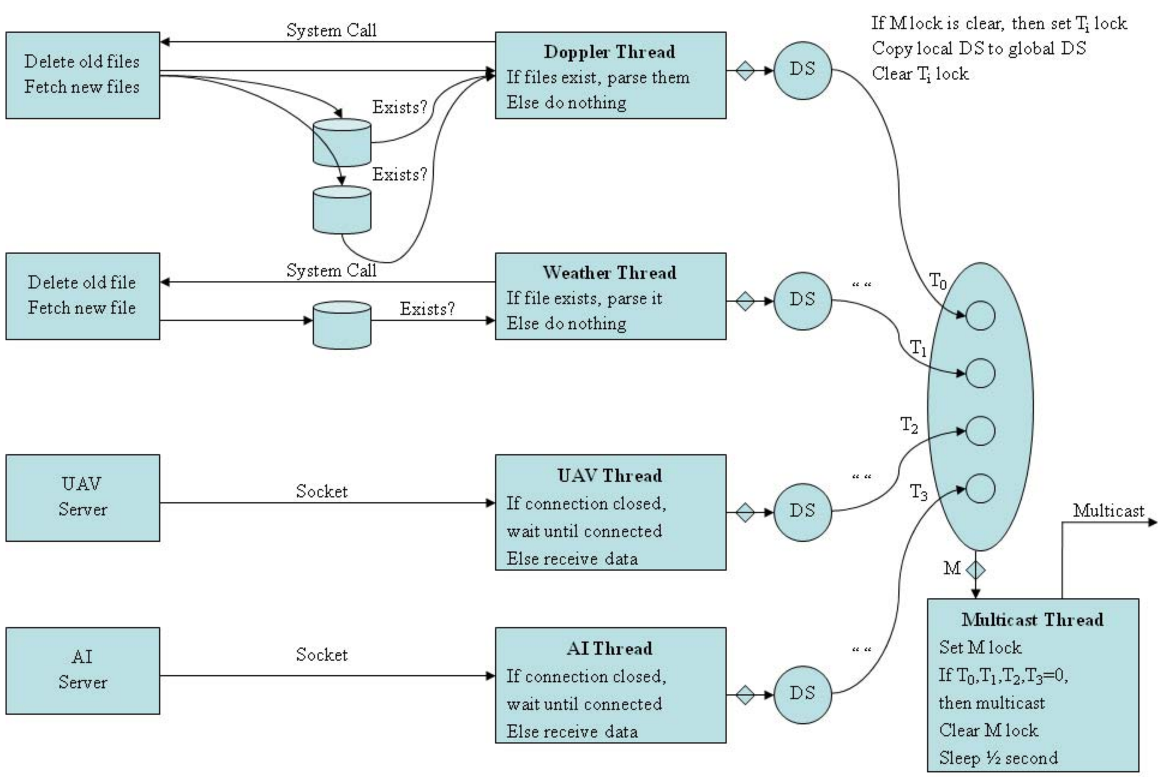

Figure 3: Data Manager architecture.

5. Conclusion

The DM is a multi-threaded software system used to assemble data from the disparate sources into a single multicast stream suitable for broadcasting to a plethora of displays that may be employed. Each of the DM's software threads samples the applicable data source at the source's Nyquist frequency. As the data is acquired it is combined into a single multicast stream, recorded on a local storage device, and multicast over a network via a multicast-enabled switch. There is a significant benefit to using a multicast approach for the delivery of information to the information display systems: Use of a multicast approach allows the system to be expandable to any number of IDSs without any architectural or software changes; allowing multiple GO IDS, which, in turn, allows for multiple UAs to operate simultaneously in our airspace. Use of a multicast approach also improves the probability that all information display systems connected to the network will be displaying the same data at any instant. This latter characteristic is of paramount importance, if the UND is to demonstrate to the FAA that UASs can operate safely in the NAS.

6. References

[1] http://www.uasresearch.org/home/default.asp?L1=1&a=29, Last accessed December, 24, 2008.

[2] Dornheim, M.A., “Flying Well With Others,” Aviation Week and Space Technology, Vol. 161, No. 5, 2004, pp. 54-56.

[3] Federal Aviation Administration, System Safety Handbook: Practices and Guidelines for Conducting System Safety Engineering and Management, December, 2000.

[4] Leveson, N., Safeware: System Safety and Computers, 1st ed., Addison-Wesley, Boston, 1995, Chapters 13-14.

[5] Federal Aviation Administration, “System Design & Analysis,” AC25.1309-1A, June 1988.

[6] Federal Aviation Administration, “Equipment, Systems, and Installation in Part 23 Airplanes,” AC23-1309-1C, March 1999.

[7] Federal Aviation Administration, Policy Memo 05-01.

[8] http://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/enroute/surveillance_broadcast/. Last accessed December, 22, 2008.

[9] http://www.faa.gov/airports_airtraffic/air_traffic/publications/bulletins/media/atb_aug_05.pdf. Last accessed December, 22, 2008.