Project Report
CBARD WXCB-1

Terminal Doppler Weather Radar (TDWR) C-Band Aerosol Release Detector (CBARD) Prototype System Design

G.R. Elkin
M.P. Matthews
S.P. Maloney
S.W. Troxel

16 December 2005

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Lexington, Massachusetts

Prepared for the Joint Program Manager for Nuclear, Biological, and Chemical Contamination Avoidance (JPM NBC CA) under Air Force Contract FA8721-05-C-0002.

ABSTRACT

The United States Joint Program Manager for Nuclear, Biological, and Chemical Contamination Avoidance (JPM NBC CA) initiated the Homeland Defense Chemical Biological Umbrella (HDCBU) program to develop, demonstrate, and deploy a capability that uses existing civil sector Doppler radars to provide an early warning of aircraft releases of chemical or biological (CB) agents. Field testing in spring 2003 resulted in the selection of the FAA Terminal Doppler Weather Radar (TDWR) for use in the development of an initial capability demonstration. During FY 2004, MIT Lincoln Laboratory led the design, development, and demonstration of a TDWR C-Band Aerosol Release Detector (CBARD) prototype system. Working closely with FAA engineers at the TDWR Program Support Facility (PSF), the CBARD prototype was tested and demonstrated to the sponsor in the summer of 2004 in Oklahoma City. In fiscal year 2005, preparations commenced for possible prototype deployment at two operational TDWR sites (Oklahoma City and Andrews Air Force Base) and a FAA command center facility in the National Capitol Region.

This document describes the CBARD system components and software design. New hardware and software running at each TDWR site allow the TDWR antenna to be scanned to detect and track an aircraft target and the plume signature associated with an aerosol release. New hardware and software at a central command center provide a human operator with an aircraft surveillance threat display (based on FAA Airport Surveillance Radar data) and also the ability to switch a TDWR into the CBARD aircraft tracking mode. If an aerosol release is detected, automated warnings are transmitted to local emergency response officials. A wide-area network provides the necessary communications infrastructure.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS
(Continued)

LIST OF TABLES

LIST OF TABLES
(Continued)

Table No. **Page**

1. OVERVIEW

1.1 PROGRAM OVERVIEW

The United States Joint Program Manager for Nuclear, Biological, and Chemical Contamination Avoidance (JPM NBC CA) initiated the Homeland Defense Chemical Biological Umbrella (HDCBU) program to develop, demonstrate, and deploy a capability that uses existing civil sector radars to provide an early warning of aircraft releases of chemical or biological (CB) agents. The premise is that early warning that such an attack has occurred would accelerate the response from the public health community. The role of a radar network would be to provide information to emergency first responders who would determine if a CB attack had occurred, and also to provide information to appropriate governmental officials.

A field measurement campaign was conducted in the spring of 2003 in Oklahoma City, in which an Environmental Protection Agency (EPA) crop duster aircraft released CB agent simulants, and several radar systems were evaluated with respect to their ability to detect the signature associated with simulant aerosol plumes. The goal of this campaign was to select a single radar to modify for an initial capability demonstration of an early warning system. In July 2003, based on an analysis of the data from these field tests, a panel of radar experts selected the Federal Aviation Administration's (FAA) Terminal Doppler Weather Radar (TDWR) as the best candidate to use in the development and deployment of the initial capability.

In fiscal year 2004, MIT Lincoln Laboratory designed, developed, and demonstrated a prototype TDWR C-Band Aerosol Release Detector (CBARD) system. Working closely with FAA engineers at the TDWR Program Support Facility (PSF), the CBARD prototype was tested and demonstrated to the sponsor in the summer of 2004 in Oklahoma City. In fiscal year 2005, preparations commenced for possible prototype deployment at two operational TDWR sites (Oklahoma City and Andrews Air Force Base) and a FAA command center facility in the National Capitol Region.

This document describes the CBARD system and software architectural design.

1.2 CBARD SYSTEM OVERVIEW

CBARD adds new hardware, software, and interfaces to allow TDWR to provide an early warning of an aerosol release from an airborne aircraft that is being tracked by a special CBARD tracking scan strategy. The primary mission of TDWR is to detect low altitude wind shear associated with thunder storms; its normal antenna scanning mode is not suitable for tracking an aircraft to support a homeland defense mission. Hence, in CBARD mode, the TDWR must be cued by an external source as to the location of a threat aircraft. To accomplish this, CBARD inputs real-time aircraft surveillance data from an FAA Airport Surveillance Radar (ASR-9). A CBARD Command Center facility provides an operator with the ability to take control of a suitably modified TDWR and activate the CBARD tracking mode. An outboard processor running at each modified TDWR supplies scan commands to the radar to allow it to track the aircraft, and detects the aircraft and aerosol plume signatures in the TDWR output reflectivity and Doppler velocity data. Automated alert messages are sent to both the Command Center and to the appropriate Emergency Operations Center (EOC) in the area where the possible agent release occurred.

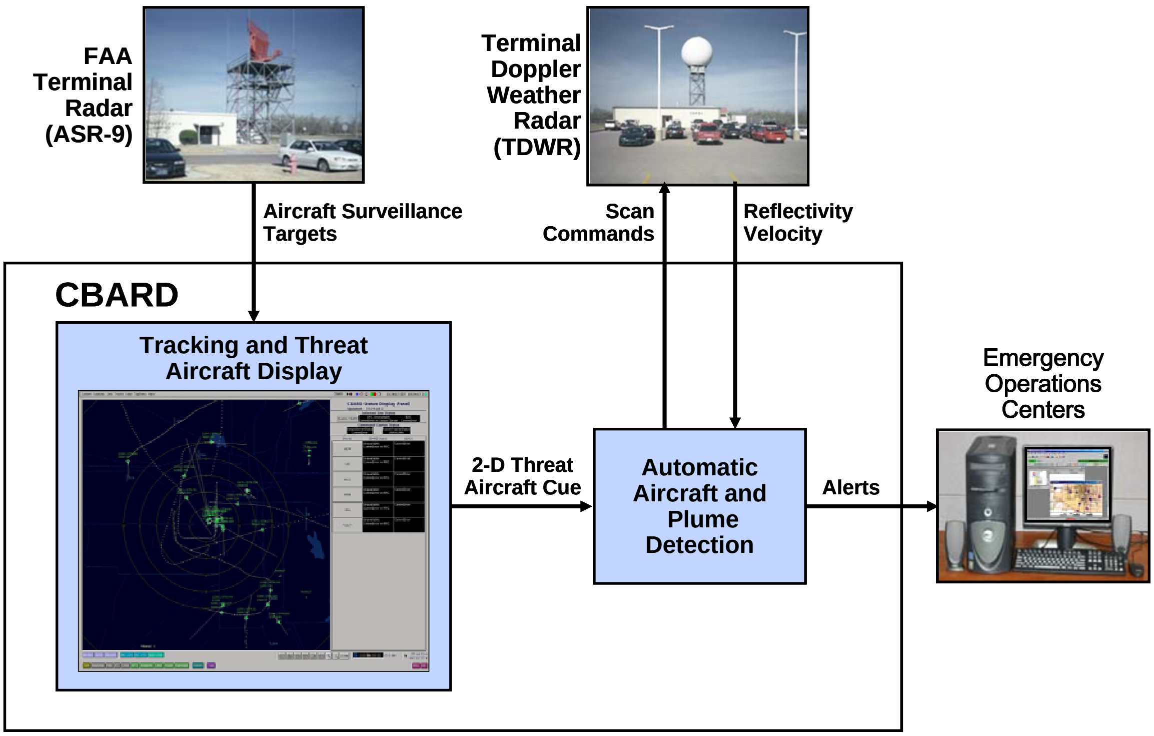

Figure 1-1. CBARD functional block diagram.

The functional block diagram shown in Figure 1-1 illustrates the major CBARD functions:

(1) Provides a human operator with an aircraft surveillance threat display;

(2) Controls the TDWR antenna in order to track aircraft targets;

(3) Detects aircraft and aerosol plume releases in the data output by TDWR; and

(4) Outputs aerosol release warnings and appropriate system status data to users.

1.2.1 Aircraft Surveillance Threat Display

The CBARD Situational Display provides an operator with situational awareness as to the location of aircraft targets throughout the CBARD coverage region. The display software is built upon a commercial Java-based software package called Tactical Display Framework (TDF), developed by Raytheon Solipsys Inc. In addition to displaying aircraft track data, the CBARD display software has been customized to provide an operator interface to send commands to any TDWR connected to the CBARD network. It also provides a system status panel that allows an operator to monitor the health of the various components of a distributed wide-area network. The plan is to install the display at the FAA WOCC.

Behind the glass, the display of aircraft track data requires a combination of computer hardware, telephone network infrastructure, and sophisticated software algorithms. The FAA radar data must be accessed and disseminated over a wide-area network. The radar target data must then be tracked to eliminate false targets and fused to provide a common, seemless surveillance picture. This last point would become important when overlapping ASR coverage occurs.

1.2.2 TDWR Aircraft Tracking Scan Strategy

The scanning mode associated with the TDWR weather detection mission is not suitable for the aircraft tracking required for the CBARD mission. Therefore, a new algorithm was developed to adaptively create and maintain an appropriate scan strategy to allow TDWR to track a single suspect aircraft. The scan strategy algorithm, which runs on a new computer situated at each TDWR facility, must continuously update the scan strategy as the aircraft moves. The algorithm has two inputs. The first input is the ASR aircraft location cues for the aircraft selected by the CBARD operator. After the TDWR executes the initial scan strategy based solely on ASR inputs, the Automated Aircraft and Plume Detection algorithm provides the second input after it detects the airplane in the TDWR reflectivity and Doppler velocity “base data.” Antenna scan strategy messages are transmitted to the TDWR Radar Product Generator (RPG) software. The RPG software was modified by FAA engineers for use in CBARD mode.

1.2.3 Automatic Detection of Aircraft and Aerosol Plumes

Normally, TDWR data processing is geared toward detection of weather phenomena, and it therefore includes algorithms to remove point targets such as aircraft from the base data. For CBARD, MIT Lincoln Laboratory developed a new algorithm to recognize the signatures in TDWR reflectivity and Doppler velocity data associated with an aircraft and aerosol plume. We will refer to this algorithm as the Aircraft and Plume Detection Algorithm. The algorithm uses an image processing technique called Functional Template Correlation (FTC) to recognize patterns in the TDWR imagery corresponding to various features associated with both the aircraft and plume. TDWR base data imagery is received via a connection to the TDWR base data Ethernet broadcast network. The algorithm also inputs ASR aircraft location cues for the selected aircraft, in order to define a search region within the TDWR imagery to mitigate false alarms. The algorithm runs on a new computer located at the TDWR facility. Aircraft detection and plume detection messages are output to CBARD Command Center for display and dissemination to non-FAA users.

1.2.4 Output of Aerosol Release Warnings

CBARD generates automated aerosol release warning messages to the appropriate Emergency Operations Center (EOC) in the area where the alleged release occurred. Each participating EOC has an HDCBU computer equipped with a NBC Analysis software package procured by the sponsor. It is the responsibility of emergency response personnel to take the appropriate actions necessary to confirm whether or not a chemical or biological agent release has occurred. The CBARD Operator Display informs the FAA operator at the CBARD Command Center that a suspected release has occurred.

1.2.5 The CBARD Network

The CBARD network architecture allows a single Command Center to control and monitor multiple TDWR facilities. As illustrated in Figure 1-2, the Command Center is connected to multiple ASR-9 data sources, multiple TDWRs, and multiple EOCs. The idea is to acquire aircraft surveillance data for each TDWR, provide two-way communications with each TDWR, and output aerosol release warnings to the appropriate emergency response facilities in each locality. The plans for implementing the CBARD network utilize two separate private wide-area networks. The radar network, whose entry points are located at FAA facilities, provides the Command Center access to aircraft surveillance data and two-way connectivity to TDWR sites. The user network connects the Command Center to EOC end user displays. This architecture was selected in order to minimize the risk of unauthorized access to TDWR computers.

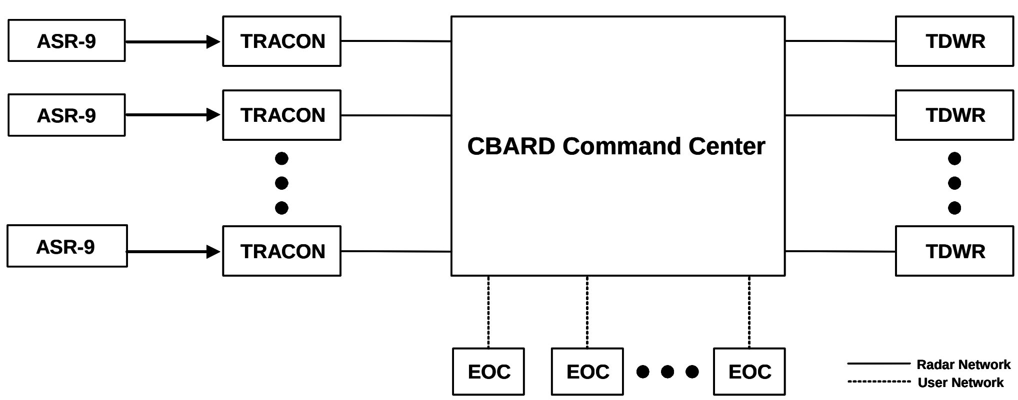

Figure 1-2. Simplified CBARD network topology.

1.3 DOCUMENT OVERVIEW

This document describes the CBARD system and software architectural design, and also discusses the program status and possible future work. The format of this document borrows some concepts from the FAA documentation standard FAA-STD-026A, a standard for documenting software development for the National Airspace System (NAS). In particular, this document combines elements of a System/Subsystem Design Description (SSDD), a Software Design Document (SDD), and an Interface Definition Document (IDD).

The contents of this document (with section numbers in parentheses) are as follows:

- Overview (1) presents a system overview and a document overview.
- System-Wide Design Decisions (2) discusses the operational concept and some key design decisions that were made.
- System Architectural Design (3) describes the allocation of system functions into Hardware Configuration Items (HWCIs), Software Configuration Items (SCIs), and manual operations. It also describes the interfaces between these Configuration Items (CIs), and the processing resources used in the prototype implementation.
- ASR Data Processing (4) describes the design of the ASR data processing software for the prototype implementation. An overview of the ASR tracking algorithms is provided.
- TDWR Data Processing (5) describes the design of the TDWR data processing software for the prototype implementation. An overview of the Aircraft and Plume Detection Algorithm is provided.
- Situational Display (6) describes the design of the CBARD situational display software for the prototype implementation.
- System Control and Monitoring (7) describes the design of the system control and monitoring software for the prototype implementation.
- Output Dissemination (8) describes the design of the software that outputs data to external users at the EOCs, as implemented in the prototype.
- Program Status and Future Work (9) discusses the program status and plans for future work.
- Appendix A provides a detailed description of all message formats used in the prototype implementation.

- Appendix B documents the results of case by case analysis of the data collected during the HDCBU Operational Demonstration test conducted in August 2004.

- Appendix C illustrates the CBARD source code directory tree structure.

A glossary of acronyms and a reference section are included at the end of the document.

2. SYSTEM-WIDE DESIGN DECISIONS

2.1 OPERATIONAL CONCEPT

The CBARD operational concept was the subject of much discussion between the JPM NBC CA and FAA. The goal was to provide a new homeland defense capability that complements existing standard operating procedures of the governmental agencies that have responsibilities for air traffic control, homeland defense, and emergency response. The FAA organized a CBARD User Working Group, consisting of FAA air traffic controllers and representatives from FAA Headquarters (many with air traffic control experience or management responsibilities). Representatives from the JPM NBC CA and MIT Lincoln Laboratory also participated in the working group. A key objective of the working group was to develop an operational concept for a specific event scenario. The scenario was exercised during the HDCBU Operational Demonstration (OD) in August, 2004, with members of the working group playing the various roles that were included in the operational concept. Representatives from the Oklahoma City EOC also participated in the OD.

The operational concept and system design were affected by a few basic procedural and programmatic constraints, as follows:

- The aggressive schedule goals imposed by the JPM NBC CA on fielding an operational prototype constrained the initial algorithm and system development effort to provide the capability to track only a single aircraft at a time at a particular TDWR site. This constraint led to the requirement for specific intelligence about a particular aircraft as the first step in the CBARD operational concept.

- Air Traffic Controllers are responsible for separation of aircraft, not homeland defense. This constraint, articulated by FAA management, led us away from assigning to controllers any responsibility for identifying a suspect aircraft or operating the CBARD Operational Display (SD). Instead, those responsibilities would be assumed by FAA personnel that operate the Domestic Events Network (DEN). Most of the key governmental organizations with interest in a CBARD-type event already have representation on the DEN, which is a telephone conference call that is staffed 24 hours a day, 7 days a week.

The working group developed the following operation concept scenario:

1. National Security or local other law enforcement agencies develop specific intelligence about a threat aircraft, and communicate this information to other governmental agencies on the DEN.

2. The FAA operator on the DEN, after consulting with appropriate governmental authorities, determines whether or not the TDWR CBARD Tracking Mode should be activated, and has the

responsibility to utilize the CBARD SD to take control of a TDWR site and activate its CBARD mode. Before activating CBARD mode, the operator must first locate the aircraft on the CBARD SD. The CBARD operator will continue to monitor the SD, in case an interruption in aircraft tracking causes CBARD mode to be deactivated, in which case the operator will re-activate CBARD mode when the aircraft track reappears.

3. After activating CBARD mode, the CBARD operator will contact the regional ATC authority and ask them to call the affected local ATC facility to request that a representative call into the DEN. The local ATC representative will then be informed that the TDWR in that locality has been placed into CBARD mode.

4. If a TDWR executing CBARD Tracking Mode detects an aerosol release from the suspect aircraft, it will transmit automated warning messages that will be displayed as alerts both on the CBARD SD and at an appropriately equipped EOC.

5. The CBARD operator, after consulting with appropriate governmental authorities on the DEN, will disable CBARD mode (using the CBARD SD), thereby returning the TDWR to its normal weather detection mission.

2.2 CBARD NETWORK DESIGN

The placement of the CBARD operator responsibility at a national command center, rather than at each FAA TRACON, led to the design of a CBARD wide-area network. The CBARD SD at a central command center has to be able to access data from and communicate with many radar and computer assets located over a wide geographical area. It must access aircraft surveillance data, establish two-way connectivity with TDWR facilities, and output warnings to EOC facilities. The CBARD network design constructs a wide-area network using private telephone lines.

The Weather Sensing Group at MIT Lincoln Laboratory has many years experience with frame relay networks, both on the Integrated Terminal Weather System (ITWS) and Corridor Integrated Weather System (CIWS) programs. A frame relay network has advantages over a dedicated leased line network. A frame relay network provides multi-point connectivity, rather than only point-to-point. However, the network service provider offers a flexible network configuration, in which the purchaser can implement only those connections that are required for the application. Frame relay networks also provide built-in alternate network paths, in case the primary path is out of service.

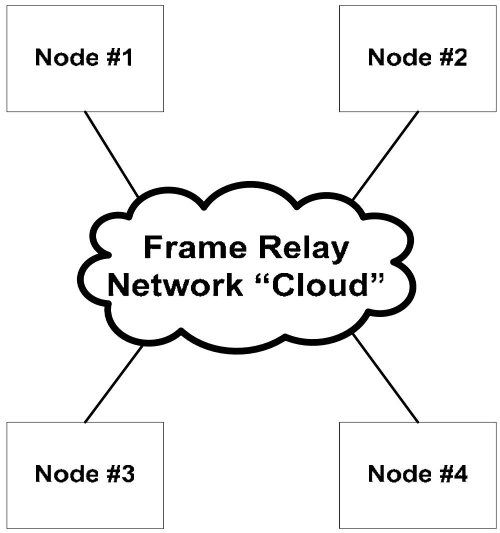

Figure 2-1. A multipoint frame relay network.

2.2.1 Network Design Requirements

The following requirements were established for the CBARD network design:

- *The system will prevent unauthorized access to TDWR equipment.* What makes the CBARD network different from ITWS and CIWS is that CBARD establishes two-way connectivity to the TDWR. The other systems receive broadcast data, but were explicitly designed so that no back channel to the radar exists. CBARD must be able to send control commands to TDWR. One way to prevent unauthorized access on a network is to set up network router hardware with access filters, which provide software based limits on network access. For CBARD, it was felt that the safest solution was to provide hardware protection instead. This is accomplished by separating the non-FAA users at the EOC from the FAA radars by serial cables, so that no network type protocol connections are possible.

- *The JPM NBC CA required that CBARD output aerosol release warnings to a Web URL at the EOC computer.* This requirement would seem to be in direct conflict with the FAA requirement for prevention of unauthorized access. Web URL traffic would have to use TCP/IP protocol, which is a two-way protocol. In order to satisfy this and the previous requirement, the external user output function was moved to a separate computer from the SD, and this output computer was placed on a separate frame relay network with connectivity to the external user (EOC) computer systems. Thus, it would be possible to use a serial link between the two networks.

- *The system will provide a status monitoring capability.* The goal was to allow the CBARD operator to monitor system status of both networks. This led to a compromise in which a two-way serial connection exists between the two networks. That allows the external network to report status back to the SD on the radar network.

- The FAA strongly urged us to provide a centralized command center capability, at which the CBARD control and monitoring equipment could be placed in the hands of the appropriate FAA personnel.

2.2.2 Network Design Alternatives

Designing the CBARD network presented many decision points where alternatives were available. One key decision was how to route CBARD output from the FAA radar network to the non-FAA EOC user network, given the requirements discussed in the previous paragraph. This paragraph presents the two designs that were considered, which we will refer to as "centralized output server" and "distributed output server", respectively.

The Centralized Output Server option, illustrated in Figure 2-2, handles all output dissemination to non-FAA users from the CBARD Command Center. With this option, aerosol release warning messages must be sent from the TDWR facility to the Command Center, and then to the appropriate EOC facilities. This was the option that was selected for the CBARD prototype implementation. The Centralized Output Server design costs less to implement and maintain. It requires less hardware, because only a single output server computer (and router, not shown in the figure) is needed. It requires fewer WAN connections, because there is only one FAA location that must be connected to the non-FAA user network. The extremely high reliability of frame relay networks mitigated the concern about routing CBARD output messages to the EOCs indirectly (via the Command Center), rather than routing them directly from the TDWR site to the EOC.

The Distributed Output Server option, illustrated in Figure 2-3, disseminates release warning data directly from each TDWR facility to the appropriate EOC facility. In order to maintain the network separation between FAA and non-FAA equipment, this option would be implemented using a separate output server computer and router at each TDWR facility. This would require added hardware cost and ongoing telephone network maintenance costs for the additional WAN connections at each TDWR site. On the other hand, this option would provide the more direct routing of CBARD outputs to the EOCs.

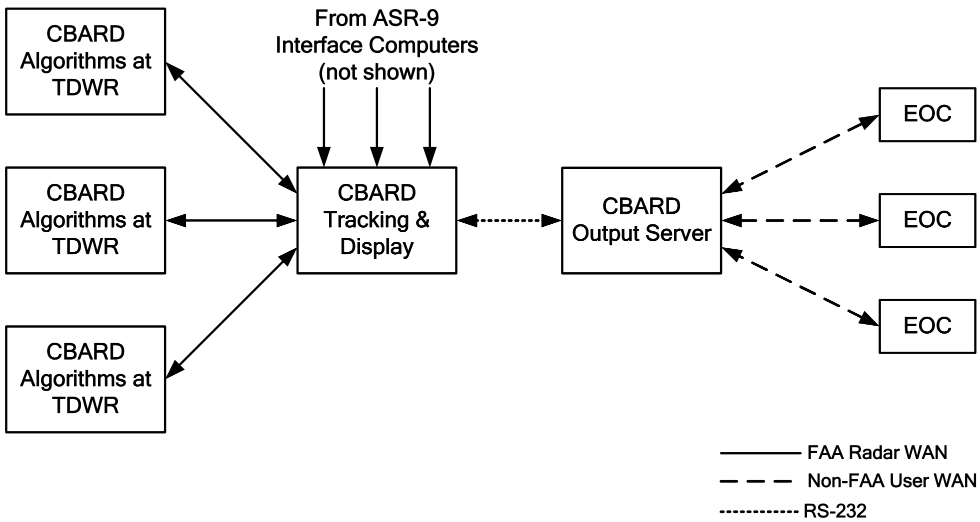

Figure 2-2. CBARD network data flow with centralized output from Command Center to EOCs.

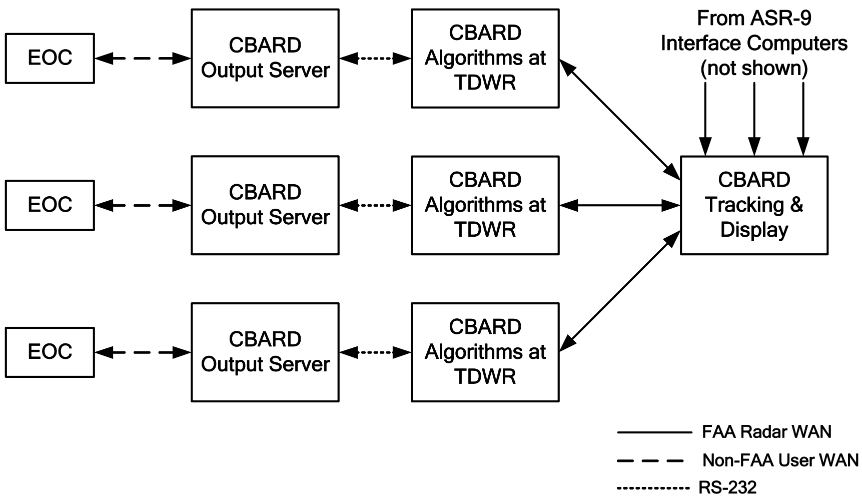

Figure 2-3. CBARD network data flow with localized output from each TDWR site to corresponding EOC.

3. SYSTEM ARCHITECTURAL DESIGN

The system architectural design section of this document identifies CBARD system components and describes the purpose of each component. It also describes the interfaces between components.

3.1 SYSTEM COMPONENTS

This section identifies each system component and describes its purpose. Each component is categorized as either HWCI, SCI, or manual operation. Table 3-1 lists the HWCIs and SCIs for CBARD.

TABLE 3-1

Hardware and Software Configuration Items

Configuration Item Identifier	Name
HWCI-TDWR	TDWR Processing Hardware
HWCI-SD	Situation Display Hardware
HWCI-ASR	ASR Processing Hardware
HWCI-OUT	Output Dissemination Hardware
HWCI-RMON	Remote Monitoring Hardware
SCI-COTS	Commercial Off the Shelf Software
SCI-SHARE	Shared Library Software
SCI-TDWR	TDWR Processing Software
SCI-ASR	ASR Processing Software
SCI-SYS	System Control and Monitoring Software
SCI-SD	Situational Display Software
SCI-OUT	Output Dissemination Software

3.1.1 HWCI Identification

Figure 3-1 presents a block diagram that illustrates the CBARD HWCIs and their interfaces.

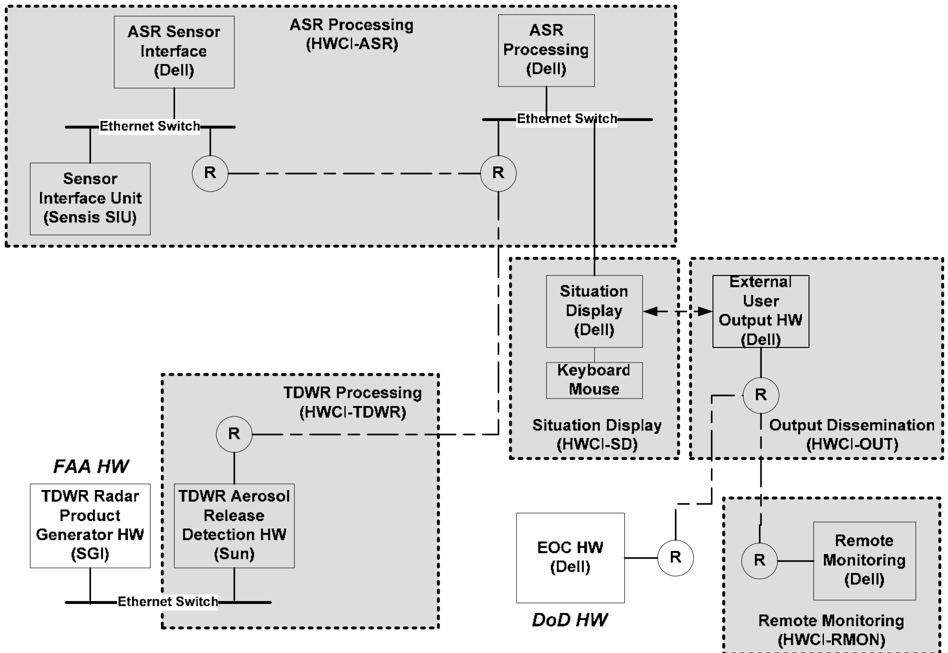

Figure 3-1. CBARD HWCI diagram.

TDWR Processing Hardware (HWCI-TDWR)

HWCI-TDWR is a computer workstation that performs the following functions:

- Executes TDWR Processing Software SCI-TDWR, which generates the CBARD aircraft tracking scan strategies for TDWR and detects the aircraft and aerosol plume release signatures in TDWR base data.

- Interfaces with the TDWR Radar Product Generator (RPG) / Remote Monitoring Subsystem (RMS) computer to exchange data relating to antenna scan strategies, threat aircraft location, system status, and plume detection.

- Interfaces with the Airport Surveillance Radar Processing hardware (HWCI-ASR) to input CBARD control commands and aircraft location cues, and output aircraft location, system status, and plume detection data.

HWCI-TDWR is a COTS computer intended for desktop, countertop, or floor use. The unit chosen for the prototype is a Sun Microsystems Inc. SunFire V250 server that includes dual SPARC processors and a DDS-4 internal tape drive. An optional flat-panel color monitor, 3-button mouse, and keyboard may be included, and can be used for system configuration and maintenance, or to display technical algorithm graphics during real-time operation. An Ethernet router (Cisco 1760) and an Ethernet switch (Cisco) provide the LAN/WAN connectivity to this HWCI. The computer is configured with three Ethernet cards, one for the TDWR Base Data network, one for the TDWR RMMS network, and one for the CBARD Command Center network.

Situational Display Hardware (HWCI-SD)

HWCI-SD, the Situation Display Hardware, is a graphics workstation that performs the following functions:

- Executes Situation Display Software SCI-SD, ASR Processing Software (SCI-ASR), and System Control and Monitoring Software (SCI-SYS).
- Interfaces to HWCI-TDWR to acquire CBARD output products and provide CBARD Operator control of TDWR CBARD mode operation.
- Interfaces to the ASR Processing Hardware (HWCI-ASR) to acquire aircraft surveillance track data.
- Displays aircraft surveillance track data graphics and system status textual data to the CBARD Operator.
- Interfaces to the Output Dissemination Hardware (HWCI-OUT) to send aerosol release warning and system status messages to external (non-FAA) users at the appropriate EOC facility. Also acquires HWCI-OUT status information.

The CBARD SD is a COTS computer and display equipment intended for desk- and/or counter-top use. The unit chosen for the prototype is a Dell Precision 360 Mini-Tower, which includes a 2.80 GHz Intel Pentium 4 processor, 2-4 GB RAM, and 1024K cache memory. A flat-panel color monitor, 3-button mouse, and keyboard are included with HWCI-SD. An Ethernet router (Cisco 1760) and an Ethernet switch (Cisco) provide the LAN/WAN connectivity to this HWCI.

ASR Processing Hardware (HWCI-ASR)

HWCI-ASR, the ASR Processing Hardware, performs the following functions:

- Executes ASR Processing Software (SCI-ASR).
- Interfaces in a read-only mode with the external hardware provided by the FAA in order to receive aircraft surveillance data.
- Interfaces with HWCI-TDWR to send CBARD control commands and aircraft location cue data, and receive aircraft location, system status, and plume detection data.

- Interfaces with HWCI-SD to output aircraft track and system status data for display and monitoring.

HWCI-ASR consists of any hardware necessary for both acquiring and processing ASR data. In the current CBARD design, HWCI-ASR includes hardware at FAA TRACONs in order to acquire ASR data, and hardware at the CBARD Command Center in order to process ASR data.

The data acquisition function currently uses a rack-mountable COTS computer, an Ethernet router and switch, and any other hardware and cables required to attach to the external ASR interfacing hardware already in place at each particular FAA facility. This equipment may include a SENSIS Sensor Interface Unit (SIU) at FAA facilities that provide ASR-9 SCIP or ASIS interface ports. The COTS computer selected for the prototype is the Dell PowerEdge 1750 rack-mountable server, which contains dual Intel Xeon 3.06 GHz processors, 512K cache memory, and 1-2 Gb RAM. The SIU-2100 was selected for use at FAA facilities where only a single ASR data feed is required.

The data processing function currently uses a single COTS rack-mountable computer connected via a local area network (LAN), including an Ethernet Router and Ethernet Switch. For the prototype, a Dell PowerEdge 1750 server is being used, which contains dual Intel Xeon 3.06 GHz processors, 512 Kb cache, 1-2 Gb RAM, and an Ethernet PCI card.

Output Dissemination Hardware (HWCI-OUT)

HWCI-OUT, the Output Dissemination Hardware, performs the following functions:

- Executes Output Dissemination Software (SCI-OUT).
- Interfaces with HWCI-SD to acquire plume detection data, and to both acquire and provide system status data.
- Interfaces to external hardware that provides an end-user situational display of aerosol plume release alerts and system status at various Emergency Operations Centers (EOCs).

The prototype implementation of HWCI-OUT uses a single COTS rack mountable Dell PowerEdge 1750 server and an Ethernet Router to connect to the user network.

Remote Monitoring Hardware (HWCI-RMON)

HWCI-RMON, the Remote Monitoring Hardware, performs the following functions:

- Executes the remote monitoring component of the System Control and Monitoring Software (SCI-SYS).
- Interfaces with HWCI-OUT to acquire plume detection and system status data.

- Provides a monitor for displaying system monitoring data, and a hard disk for logging data.

The prototype implementation of HWCI-RMON uses a Dell Precision 530 tower computer connected to the user network.

3.1.2 SCI Identification

The CBARD system design includes commercial off-the-shelf (COTS) software such as the POSIX compliant UNIX operating system variants (Linux, Solaris). Because such software is not considered a CBARD developmental item, it is not addressed in this document. The developmental SCIs, illustrated in the block diagram in Figure 3-2, are described in the subsections below.

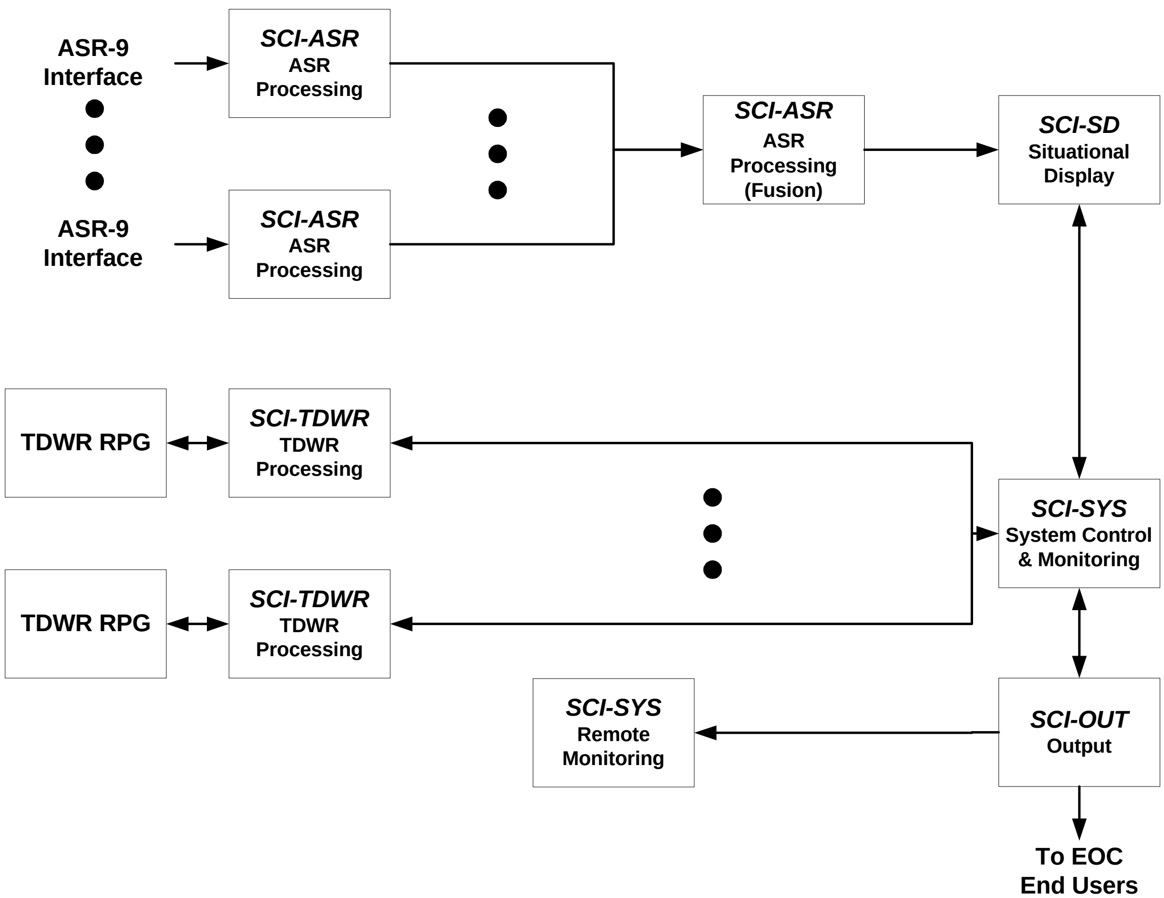

Figure 3-2. CBARD SCI diagram.

Shared Library Software (SCI-SHARE)

The Shared Library Software consists of libraries shared by many of the other SCIs to perform interprocess communications, recording of algorithm products and messages, and logging of diagnostic data. Table 3-2 summarizes the functions performed by SCI-SHARE within other SCIs. The shared software libraries used by CBARD are SCLite, WxObject, and Log. These libraries will not be documented in this report. Documentation of SCLite and Log can be found in [9].

TABLE 3-2

Functions Performed by SCI-SHARE within Other SCIs

SCI	Function
SCI-TDWR	Transfers data to/from external interface to TDWR Radar Product Generator (RPG). Transfers data to and from System Control and Monitoring Software (SCI-SYS). Records stored algorithm product data and diagnostic data to disk.
SCI-ASR	Transfers data to and from System Control and Monitoring Software (SCI-SYS). Records diagnostic data to disk.
SCI-SYS	Transfers data to and from ASR Processing Software (SCI-ASR). Records diagnostic data to disk.
SCI-OUT	Transfers data among internal software processes within SCI-OUT. Transfers data to the Remote Monitoring software component of SCI-SYS. Records diagnostic data to disk.

TDWR Processing Software (SCI-TDWR)

The TDWR Processing Software performs the following functions within HWCI-TDWR:

- Uses the automated aircraft and plume detection algorithm to generate the CBARD aircraft tracking scan strategies for TDWR based on input aircraft location cues from SCI-SYS and TDWR base data from the TDWR RPG.
- Detects the aircraft and aerosol plume release signatures in the TDWR base data.
- Outputs aircraft location, plume detection, and status data to SCI-SYS.

ASR Processing Software (SCI-ASR)

The ASR Processing Software performs the following functions within HWCI-ASR:

- Inputs ASR-9 sensor target data by connecting to an FAA ASR-9 interface port or existing network with access to such data.

- Sends ASR data to the CBARD Command Center using TCP/IP protocol over a network.

- Performs tracking of single-sensor ASR data, and fuses tracks from multiple sensors.

- Outputs aircraft surveillance track data to the Situational Display Software, SCI-OUT.

System Control and Monitoring Software (SCI-SYS)

The System Control and Monitoring Software performs the following functions within HWCI-SD:

- Receives system control messages from SCI-SD resulting from CBARD operator actions, and forwards these messages to the appropriate SCI-TDWR at the appropriate CBARD-modified TDWR site.

- Receives plume detection messages, TDWR aircraft location messages, and system status messages from all TDWR-based SCI-TDWR subsystems, and forwards these messages to SCI-SD and SCI-OUT, as necessary.

- Receives status messages from SCI-OUT, and forwards these messages to SCI-SD.

The System Control and Monitoring Software perform the following functions within HWCI-RMON:

- Receives system status and plume detection messages from SCI-OUT.

- Displays and logs to disk received messages to facilitate remote system monitoring.

Situational Display Software (SCI-SD)

The Situational Display software performs the following functions within HWCI-SD:

- Provides graphical user interface to allow the CBARD Operator to control and monitor the CBARD mission.

- Interfaces with SCI-ASR to acquire aircraft track data to be displayed.

- Interfaces with SCI-SYS to send CBARD control commands and threat aircraft cue data to the TDWR, and to receive plume detection, aircraft location, and system status data from the TDWR.

Output Dissemination Software (SCI-OUT)

The Output Dissemination Software performs the following functions within HWCI-OUT:

- Receives plume detection and system status messages from SCI-SYS.
- Translates plume detection and system status messages into the external user NBC-1 message format.
- Sends aerosol release warning messages to the appropriate EOC external user systems.
- Sends status messages to SCI-SYS.
- Forwards received system status and plume detection messages to the remote monitoring component of SCI-SYS.

3.1.3 Manual Operations

Manual operations are those that are required to be performed by maintenance personnel, users, or other operators in order for the system to fulfill requirements; they do not include ordinary operator interactions with the CBARD SD Computer Human Interface (CHI). For CBARD prototype operations, the key manual operation is to load/unload transportable electronic media in support of software upgrades and data archival. For the prototype phase, this is solely the responsibility of MIT Lincoln Laboratory personnel. The JPM NBC CA has developed a plan to conduct quarterly tests of the CBARD capability. That plan requires that recorded data and log files be retrieved on transportable electronic media at that time. In addition, a requirement would likely be specified for the handling of data in the event of an actual use of CBARD if a WMD aerosol release occurred.

All recorded CBARD data can be captured from the Command Center via the CBARD radar network, including TDWR Base Data, CBARD aerosol release detection algorithm products, ASR-9 aircraft surveillance data, and system status data. TDWR-based data can also be archived at each TDWR site using the 4mm DAT drive attached to the CBARD algorithm computer. FAA also has procedures in place to allow its radar site technicians to capture Base Data recordings at the TDWR site.

3.1.4 Relationships Among HWCIs and SCIs

Figure 3-3 illustrates the relationships between the CBARD system components, HWCIs, and SCIs. If deployed, CBARD hardware will be distributed across the United States. The TDWR Processing Hardware, HWCI-TDWR, will be located at each TDWR facility that is equipped to perform the CBARD mode function. HWCI-TDWR executes SCI-TDWR.

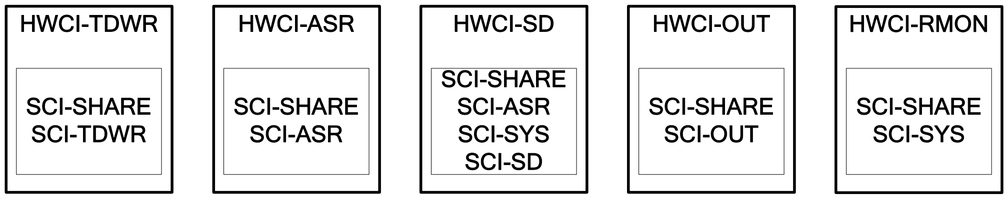

Figure 3-3. Relationships between CBARD components, HWCIs, and SCIs.

HWCI-ASR will be partially located at each TRACON to provide an interface for accessing an ASR whose coverage region best matches a TDWR that is equipped to perform the CBARD mode function. Other HWCI-ASR components will be located at the CBARD Command Center, in order to process ASR data. HWCI-ASR executes SCI-ASR.

HWCI-SD and HWCI-OUT will be located at the CBARD Command Center. HWCI-SD executes SCI-ASR, SCI-SYS, and SCI-SD. HWCI-OUT executes SCI-OUT.

For the CBARD prototype phase, HWCI-RMON will be located at MIT Lincoln Laboratory. HWCI-RMON executes one component of SCI-SYS.

3.2 INTERFACE DESIGN

This section identifies all CBARD external and internal interfaces, and provides a high-level description of each interface. Message format details are presented in Appendix A. Software interfaces are discussed in more detail in the chapters 4-8, each of which describes a particular SCI and its interfaces.

3.2.1 Interface Identification and Diagrams

Figure 3-1 illustrates the interfaces between HWCIs. Figure 3-2 illustrates the interfaces between SCIs.

3.2.2 ASR-9 : HWCI-ASR Interface

HWCI-ASR interfaces with the ASR-9 Remote SCIP via serial RS-232 communication lines at an FAA TRACON. This interface allows CBARD to acquire ASR-9 target channel data in real-time.

3.2.3 HWCI-TDWR : HWCI-ASR Interface

HWCI-TDWR interfaces with HWCI-ASR via TCP/IP on a private wide-area Ethernet network. This interface passes CBARD control data from the Command Center to the TDWR. It also passes CBARD algorithm outputs and system status data from the TDWR site to the Command Center.

3.2.4 HWCI-ASR : HWCI-SD Interface

HWCI-ASR interfaces with HWCI-SD via TCP/IP and UDP/IP on a private local-area Ethernet network. This interface passes aircraft surveillance track data to the SD.

3.2.5 HWCI-SD : HWCI-OUT Interface

HWCI-SD interfaces with HWCI-OUT via a two-way serial RS-232 communications link. This interface passes aerosol release detection data, TDWR-based aircraft detection data, and system status data to the CBARD Output Server computer.

3.2.6 HWCI-OUT : HWCI-RMON Interface

HWCI-OUT interfaces with HWCI-RMON via TCP/IP on a private wide-area Ethernet network. This interface passes CBARD algorithmic outputs and system status data to a remote monitoring computer (to be located at MIT Lincoln Laboratory during the prototype phase).

3.2.7 SCI-TDWR : HWCI-TDWR Interface

SCI-TDWR writes algorithm product data and software log data onto the HWCI-TDWR computer hard disk drive.

3.2.8 SCI-ASR : HWCI-ASR Interface

SCI-ASR writes aircraft surveillance data and software log data onto the HWCI-ASR computer hard disk drive.

3.2.9 SCI-ASR : HWCI-SD Interface

SCI-ASR writes aircraft surveillance track data and software log data onto the HWCI-SD computer hard disk drive.

3.2.10 SCI-SYS : HWCI-SD Interface

SCI-SYS writes software log data onto the HWCI-SD computer hard disk drive.

3.2.11 SCI-SYS : HWCI-RMON Interface

SCI-SYS writes software log data onto the HWCI-RMON computer hard disk drive.

3.2.12 SCI-SD : HWCI-SD Interface

SCI-SD writes display software log data onto the HWCI-SD computer hard disk drive.

3.2.13 SCI-OUT : HWCI-OUT Interface

SCI-OUT writes aerosol release warning data and software log data onto the HWCI-OUT computer hard disk drive.

3.2.14 SCI-TDWR : SCI-SYS Interface

A two-way interface between SCI-TDWR and SCI-SYS provides for the transfer of system control data from SCI-SYS to SCI-TDWR, and also the transfer of algorithm results and system status from SCI-TDWR to SCI-SYS. The specific message types are discussed in Sections 5 and 7, and detailed message formats are presented in Appendix A. Communication between SCI-TDWR and SCI-SYS will be accomplished via a private wide-area network, since SCI-TDWR runs within HWCI-TDWR at each TDWR facility and SCI-SYS runs within HWCI-SD at the Command Center. A 56 kbps link will provide sufficient capacity. The TCP/IP network protocol will be used for all messages sent over this interface. SCI-SHARE provides software library subroutines (SCLite and WxObj) for use in sending and receiving messages using TCP/IP.

3.2.15 SCI-TDWR : TDWR RPG Interface

SCI-TDWR interfaces with the TDWR RPG software via TCP/IP and UDP/IP on an Ethernet LAN. This interface passes CBARD control data and aircraft location cueing data to the RPG via TCP/IP in order to execute CBARD surveillance mode. The RPG passes TDWR Base Data via UDP/IP broadcast to the CBARD aircraft and plume detection algorithms running on SCI-TDWR. This interface also passes aircraft detection data and system status data between the two SCIs via TCP/IP.

3.2.16 SCI-ASR : SCI-SYS Interface

SCI-ASR interfaces with SCI-SYS via TCP/IP on an Ethernet LAN at the CBARD Command Center. This interface passes aircraft location cue data from SCI-ASR to SCI-SYS (for eventual relay to SCI-TDWR). The specific message types are discussed in Sections 4 and 7, and detailed message formats are presented in Appendix A. SCI-SHARE provides software library subroutines (SCLite and WxObj) for use in sending and receiving messages using TCP/IP.

3.2.17 SCI-ASR : SCI-SD Interface

SCI-ASR interfaces with SCI-SD via UDP/IP broadcast mode on an Ethernet LAN at the CBARD Command Center. This interface passes ASR-9 aircraft track data to the SD. Further details are provided in Sections 4 and 6.

3.2.18 SCI-SYS : SCI-SD Interface

SCI-SYS interfaces with SCI-SD via TCP/IP on an Ethernet LAN at the CBARD Command Center. This interface passes algorithm aerosol release detection, TDWR-based aircraft location, and system status data to the SD. The specific message types are discussed in Sections 6 and 7, and detailed message formats are presented in Appendix A.

3.2.19 SCI-SYS : SCI-OUT Interface

SCI-SYS interfaces with SCI-OUT in order to pass CBARD system status and aerosol release detection data from SCI-SYS to SCI-OUT, and also to pass system status data from SCI-OUT to SCI-SYS. The specific message types are discussed in Sections 7 and 8, and detailed message formats are provided in Appendix A. Communication between SCI-SYS and SCI-OUT will be accomplished via a serial RS-232 cable connection. SCI-SYS and SCI-OUT execute within computer hardware located in the CBARD Command Center.

3.2.20 SCI-SD : CBARD Operator Interface

SCI-SD interfaces with the CBARD Operator via the SCI-SD Computer Human Interface (CHI). Section 6 discusses the design and some of the features of the CBARD SD CHI.

3.2.21 SCI-OUT : EOC Interface

SCI-OUT interfaces with the NBCAnalysis software executing on the HDCBU EOC computer (external to CBARD equipment). SCI-OUT interfaces with the EOC NBCAnalysis software via TCP/IP on an Ethernet WAN. This interface passes NBC-1 formatted aerosol release warning and CBARD mission status messages to the NBCAnalysis software. The message format details are provided in Appendix A.

4. ASR-9 DATA PROCESSING (SCI-ASR)

4.1 OVERVIEW

This section describes the software design for the CBARD ASR-9 Data Processing Software (SCI-ASR). SCI-ASR performs the processing necessary to acquire ASR-9 aircraft surveillance data and generate track data that can be presented to the CBARD Operator. The focus will be on the software architectural design, including the software components and their interfaces. This overview discusses ASR-9 data and how it is acquired, describes at a high level the track generation algorithms, and introduces two software libraries developed at MIT Lincoln Laboratory for other projects that were borrowed for the CBARD prototype implementation of SCI-ASR.

4.1.1 ASR-9 Interface

With FAA approval, ASR-9 aircraft surveillance data can be acquired via a hardware connection to interface ports located either at the radar facility or the corresponding Terminal Radar Control (TRACON) facility, as illustrated in Figure 4-1. HWCI-ASR (discussed in section 3.1.1) consists of the hardware necessary to access ASR-9 data, including serial cables, a serial-to-Ethernet protocol converter, and a network router and wide-area network. CBARD employs a software library called the Surveillance Data Network (SDN) Software Development Toolkit (SDK). The radar data is transmitted over a wide-area network to the Command Center facility where the track data generation function is executed.

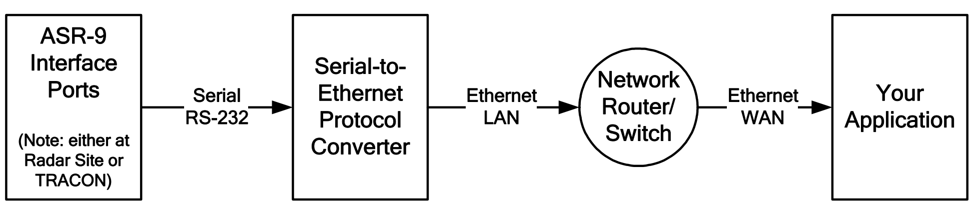

Figure 4-1. Typical interface to access ASR-9 data.

4.1.2 ASR-9 Radar and Beacon Processing

In order to understand the SCI-ASR track data generation function, it is helpful to understand the type of aircraft target data output by the ASR-9. All of the busiest airports in the United Status are equipped with two co-located aircraft surveillance radar systems: a primary (a.k.a., "skin") radar, such as the ASR-9, and secondary (a.k.a. beacon) surveillance radar, such as Mode S. The beacon system provides active surveillance of transponder-equipped aircraft. The ASR-9 provides passive surveillance of aircraft, whether or not they carry a transponder. It should be noted that many small general aviation aircraft, such as crop dusters, operating in rural areas are not transponder equipped. As shown in Figure 4-2, the ASR-9/Mode S system outputs an integrated primary/beacon target data stream, called "uncorrelated" data. It also has a scan-to-scan correlator algorithm that acts as a filter to generate a delayed data stream called "correlated" that is a subset of the uncorrelated stream. Correlated data is the result of a tracking algorithm that lives within the ASR-9. Unfortunately, for CBARD's purposes, the correlated data output does not contain any beacon targets, nor does it disseminate a track number for each correlated target. Therefore, CBARD must use the uncorrelated data and perform its own tracking function in order to generate an integrated aircraft surveillance picture for the CBARD operator.

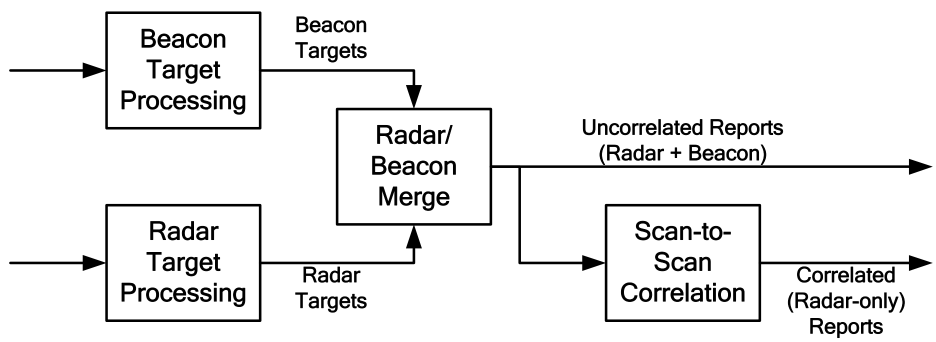

Figure 4-2. ASR-9/Mode S data flow diagram.

4.1.3 ASR Aircraft Tracking Algorithms

In order to provide wide-area aircraft surveillance coverage, the CBARD aircraft tracking function consists of two stages of processing: single-sensor tracking and track fusion, as illustrated in Figure 4-3. Each ASR-9 surveillance output stream is processed by a separate instance of the single-sensor tracker, each of which outputs a local track report data stream. A single track fusion process inputs all of the local track report data streams, correlates the input to generate a fused global track file, and then outputs the global track report data to the CBARD SD. Where there is overlapping coverage from more than one radar, the fusion tracker will maintain only a single track for each aircraft.

An overview of the single-sensor and fusion tracking algorithms are provided in the following paragraphs.

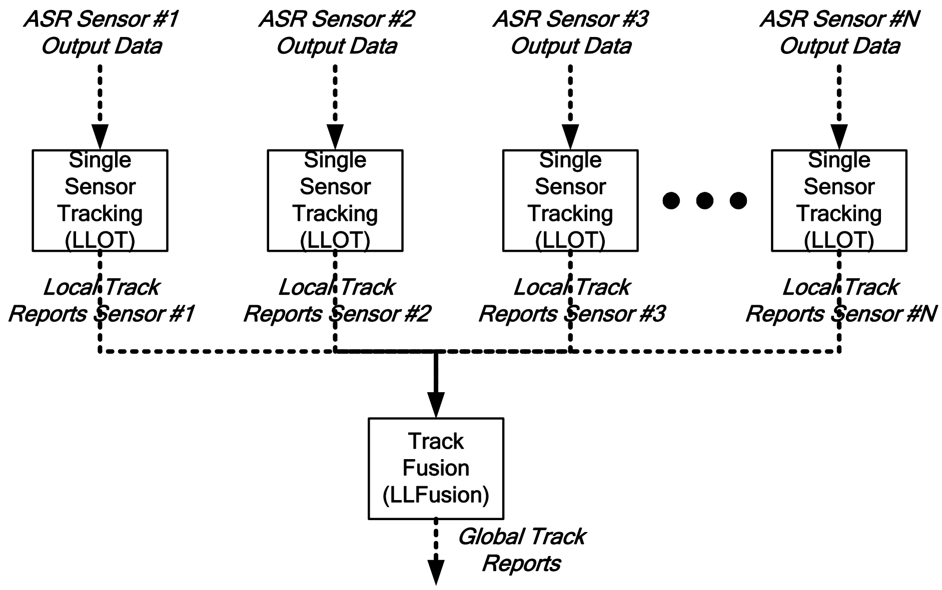

Figure 4-3. CBARD ASR aircraft tracking algorithm stages and corresponding data flow.

Single-Sensor Tracking Algorithm

The CBARD single-sensor tracking algorithm is based the tracking algorithms that are currently used by the ASR-9 and Mode S sensors. The most recent version of the ASR-9 tracker was developed at MIT Lincoln Laboratory for the ASR-9 Processor Augmentation Card (9PAC) program during the 1990's. CBARD uses an updated version of the 9PAC tracking software that was developed recently in a FAA funded effort to re-host the ASR-9/ModeS surveillance functions to a new hardware platform. In this document, we will refer to this tracker as the Lincoln Laboratory Optimized Tracker (LLOT). Although there is no published documentation on the LLOT version of the ASR-9/Mode S tracking algorithms, algorithm details are available for the 9PAC tracking algorithm on which LLOT is based [3]. LLOT beacon tracking algorithms are documented in the Mode S specification [4].

Fusion Tracking Algorithm

CBARD uses a fusion tracker recently developed at MIT Lincoln Laboratory called LLFusion. Since the LLFusion algorithm has not been documented to date, a high-level overview of its processing steps is provided below. The basic idea is to convert incoming single-sensor track data into a common global coordinate system, and then look for track matches where overlapping coverage exists. Track matching criteria include position and beacon code and altitude data when available [6].

A configuration file specifies the name and location (LAT/LON) of each sensor providing data, along with the reference coordinate for a global stereographic coordinate system [10]. Each received local track report includes a sensor number, local track number, track position (range and azimuth) relative to the sensor, time of measurement, and beacon code and altitude if available. For each input, LLFusion converts the track position to the global (x,y,z) coordinate system. A local track file is maintained for each sensor. The first local track report for a particular track number (from a particular sensor) initiates a new local track file entry. Each local track file entry maintains smoothed speed and heading estimates for the track as new inputs are received for the same track.

The global track association algorithm groups local tracks from different sensors into a single global track file, such that each aircraft is associated to a single global track entry. Once a local-to-global track assignment has been made, subsequent updates to the local track will be compared against the global track to determine if the association should be continued. Criteria used for comparing local to global tracks includes beacon code and altitude (if present), and nearness of track positions. Global track parameters are updated based on the newly associated local track report. The speed and heading estimates are computed using a weighted average of the global track and its new associate. If a local track fails to match its global track for three consecutive updates, the local and global tracks are disassociated. A disassociated local track may be able to match a different global track. The algorithm allows a local track to establish a “trial track” match with another global track. In order to switch to the other global track, the local track must have the same beacon code and altitude, or else match the position and velocity of the global track, or else match the position of the global track for 3 consecutive inputs. If a local track fails to match any existing global track, a new global track is initiated.

LLFusion outputs a global track report for each input local track report. The output includes the track number (called a designated track number, or DTN), location (global x,y,z and Lat/Lon), speed and heading, and the beacon code and altitude if available.

4.1.4 Surveillance Data Network (SDN) Software

SDN provides a distributed, open systems approach for surveillance data management and dissemination. Developed at MIT Lincoln Laboratory under FAA sponsorship, SDN provides a means of delivering an integrated air picture in an environment in with numerous and diverse sensors exist, and there is a need for information sharing. The ultimate goal of SDN is to provide a wide-area network on which sensor data can be ingested by various surveillance applications, which then generate and publish refined products that can be subscribed to by other applications. This publish-subscribe model of information sharing can be contrasted with the point-to-point model typically used in FAA surveillance systems to date. The SDN program at MIT Lincoln Laboratory maintains a network that is connected to many sensors across the Continental United States (CONUS). SDN concepts are currently in use in applications ranging from Air Traffic Control to Ballistic Missile Defense.

For the CBARD prototype, we were not able to connect to the existing SDN network and simply subscribe to existing tracking products, because SDN was not ingesting Oklahoma City ASR-9 (OKC) data. Therefore, while CBARD was designed to use the SDN Software Development Kit (SDK) to provide reliable ingest of ASR-9, a simple point-to-point model of information sharing was adopted. As shown in Figure 4-4, CBARD uses a pair of SDN software tasks for each ASR-9 radar on the CBARD network. The Sensor Interface (SI) task ingests ASR-9 from the appropriate interface point, and sends this data over a network to the corresponding Sensor Manager (SM) task.

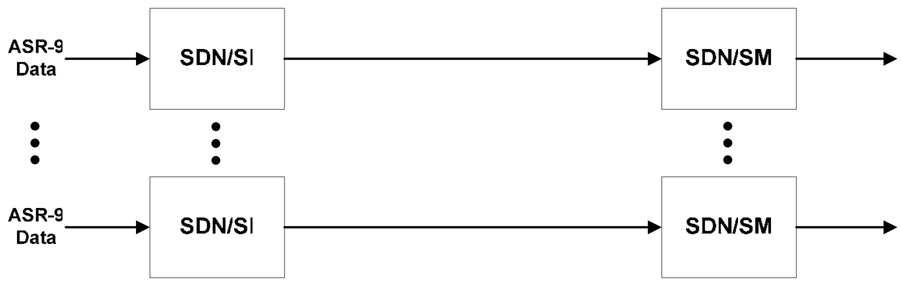

Figure 4-4. CBARD uses SDN Sensor Interface (SI) and Sensor Manager (SM) software for each ASR-9 data stream ingested.

At the time that CBARD development was ongoing, the SDN development team was in the process of adding the ASR-9 tracking and track fusion software to its SDK product suite. Therefore, in order to get the CBARD prototype working quickly, the SSTB software discussed in the following paragraphs was used for the track generation functions, because a working version of the tracking software was already available. If CBARD were to be fielded, it would be appropriate to eliminate the SSTB software entirely and just use the SDN SDK.

4.1.5 Surveillance System Test Bed (SSTB) Software

Developed at MIT Lincoln Laboratory, the Surveillance System Test Bed (SSTB) software provides algorithms for acquiring and processing data from several radar systems, including the ASR-9. SSTB is built upon the TaskRunner software engine [5], which provides an application programmer interface (API) for developing real-time system software that is independent of the operational target hardware and operating system. The TaskRunner API provides a mechanism for attaching various software tasks for processing radar data. Using a simple text script file, a series of software tasks can be chained together, such that outputs from one task can be input by other tasks.

The SSTB Software Development Kit (SDK) version 5.6 was used for CBARD prototype development. Enhancements to SSTB 5.6 developed for CBARD include hardening of the interface to the single-sensor ASR-9 tracker (LLOT), and a new task that provides a cueing message containing the data associated with the track "selected" on the CBARD situational display. CBARD uses a separate TaskRunner script for each incoming ASR-9 data stream (the SSTB Sensor Tracker), and a single TaskRunner script for the track fusion and TDWR cueing tasks (the SSTB Fusion Tracker). The SSTB Fusion Tracker interfaces with the CBARD Situational Display software (SCI-SD) to output track data for display and receive data on operator actions in order to generate the CBARD enable, disable, and aircraft cue messages that allow a TDWR to execute CBARD mode.

4.2 ARCHITECTURAL DESIGN

4.2.1 SCI-ASR Components

The two main software components of SCI-ASR are SDN and SSTB, each of which is a software library that consists of many software units. We will describe the major software units used by CBARD. Figure 4-5 illustrates the "consists of" relationships among the software units. SDN consists of a Sensor Interface (SI) unit, a Sensor Manager (SM) unit, and a Naming Service unit. SSTB consists of a Sensor Tracker unit and a Fusion Tracker unit. The SSTB Sensor Tracker and Fusion Tracker units are each composed of individual software tasks. These are discussed in the following paragraphs.

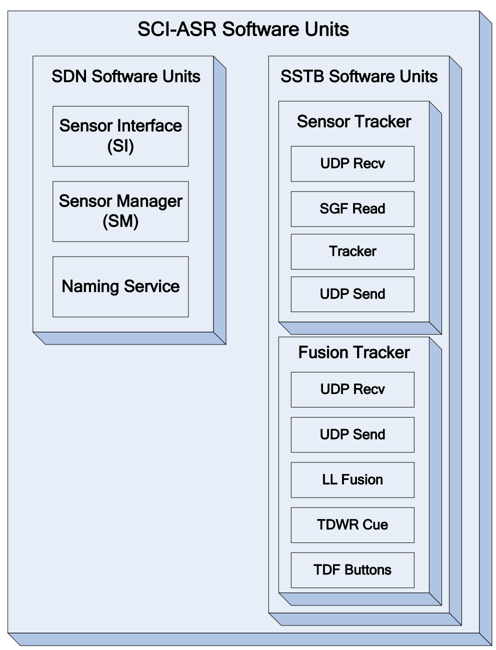

Figure 4-5. SCI-ASR software units.

SDN Sensor Interface (SI)

The SDN SI performs the following functions within SCI-ASR:

- Inputs ASR-9 target data from an Ethernet UDP broadcast stream.
- Puts TCP/IP and CORBA wrappers around the target data, and outputs the data to the SDN SM application using the TCP/IP protocol over a private wide-area network.

SDN Sensor Manager (SM)

The SDN SM performs the following functions within SCI-ASR:

- Inputs ASR-9 target data from the UDP broadcast stream output by the SDN SI.
- Removes the CORBA and TCP/IP wrappers, and then outputs the target data to SCI-SYS using UDP broadcast protocol.

SDN Naming Service

The SDN Naming Service binds together each pair of SI and SM tasks, so they can communicate over a network. CBARD uses a single Naming Service task that provides this service to all SI/SM pairs. However, it is possible to build a "federation naming service", so that failure of one will not bring down the whole system [7].

SSTB Sensor Tracker

The SSTB Sensor Tracker performs the following functions within SCI-ASR:

- Inputs ASR-9 target data from the UDP stream output by the SDN SM.
- Executes the LLOT algorithm.
- Outputs a single-sensor track data stream using UDP broadcast protocol.

The SSTB Sensor Tracker is a Java process made up of several software tasks, as shown in Figure 4-6. A separate instance of the Sensor Tracker process runs for each incoming ASR-9 data stream. Each ASR-9 data stream is broadcast on a unique UDP socket. The udp_recv task attaches to the appropriate socket, reads the ASR-9 data, and forwards it to the SGF_Read task using the SSTB TaskRunner "output" method. SGF_Read is a task that parses the SGF format wrappers and forwards the underlying ASR-9 target reports to the LLOT task. LLOT runs the single-sensor tracking algorithm, and outputs track reports to the udp_send task. The udp_send task sends the track report data to the SSTB Fusion process on another UDP socket stream.

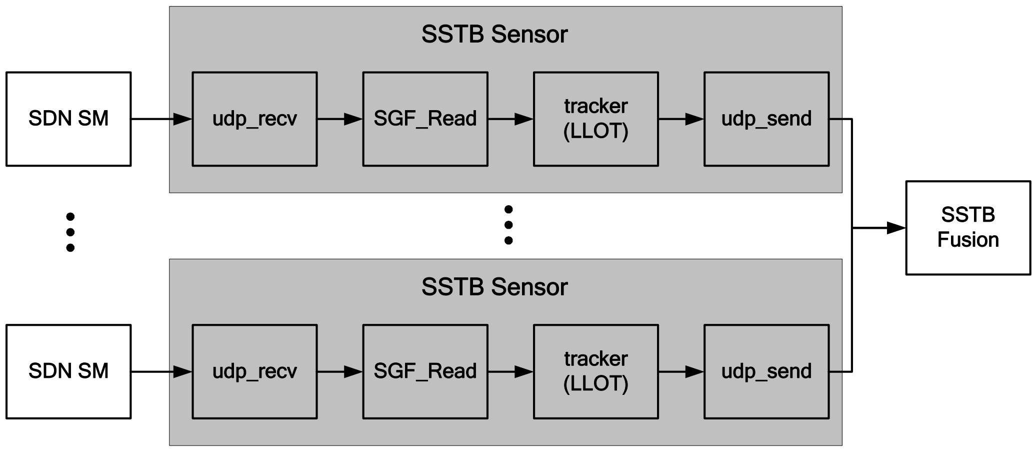

Figure 4-6. SSTB sensor tracker software.

SSTB Fusion Tracker

The SSTB Fusion Tracker performs the following functions within SCI-ASR:

- Inputs single-sensor track data from each single-sensor track data UDP broadcast stream.
- Executes the LLFusion algorithm.
- Outputs a single fused track data stream using a UDP broadcast protocol.
- Inputs messages indicating actions taken by the CBARD Operator using the SD (SCI-SD), in particular including the selected designated track number (DTN) and commands to activate or deactivate CBARD Tracking Mode at a particular TDWR site.
- Outputs aircraft location cues and CBARD Track Mode activation/deactivation commands to the System Control and Monitoring Software (SCI-SYS), in particular to its TDWR Server software component.

The SSTB Fusion Tracker is a Java process made up of several software tasks, as shown in Figure 4-7. The SSTB Fusion process has a separate udp_recv task to listen to each UDP socket broadcast stream coming from the individual SSTB Sensor Tracker processes. Each udp_recv task uses the SSTB TaskRunner output method to forward the incoming sensor track data to the LLFusion task. LLFusion executes the fusion tracking algorithm, and outputs the resulting single fused track data stream to two tasks, tdf_dump and TDWRCue. The tdf_dump task forwards the fused track data to tdf_send, which forwards the data to udp_send. The udp_send task sets up another UDP broadcast socket to send the fused track data to the CBARD SD (SCI-SD). When the CBARD Operator selects a track or issues a CBARD mode command, SCI-SD sends a UDP broadcast that is received by another udp_recv task, which forwards these data to the TDFButtons task. TDFButtons filters through the Operator action data stream, and outputs the track selections and CBARD mode commands to TDWRCue, which is also receiving the fused track data from LLFusion. This allows TDWRCue to look up the track data that corresponds to the track selected by the operator, and then format and send CBAircraftLoc, CBEnable, and CBDisable messages to the SCI-SYS TDWRServer task (which communicates with the TDWR sites).

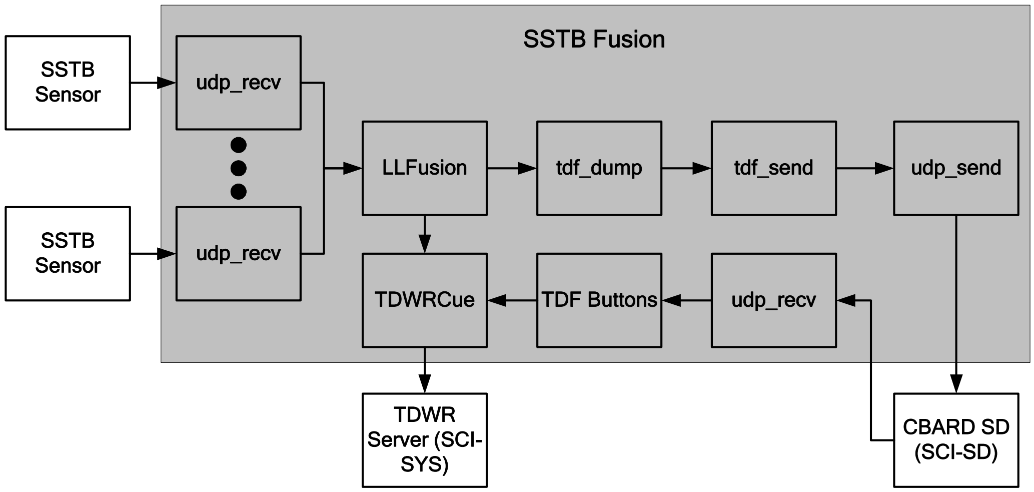

Figure 4-7. SSTB fusion tracker software.

4.2.2 Interface Design

Interface Identification and Diagrams

Figure 4-8 illustrates the software interfaces for SCI-ASR major functional components. The software interfaces include SIU:SI, SI:SM, SM:SensorTracker, SensorTracker:FusionTracker, and FusionTracker:SD. In addition, there is an interface between each SI and the Naming Service (SI:NamingService), and between each SM and the Naming Service (SM:NamingService).

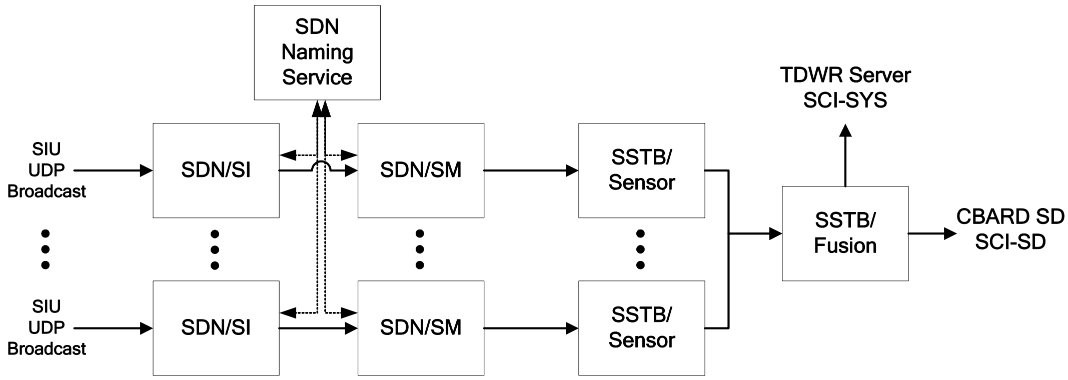

Figure 4-8. SCI-ASR software interfaces.

Figure 4-9 illustrates the SDN software interfaces, including the SI:SM interface, the SI:NamingService interface, and the SM:NamingService interface. The SM:SensorTracker interface is also shown.

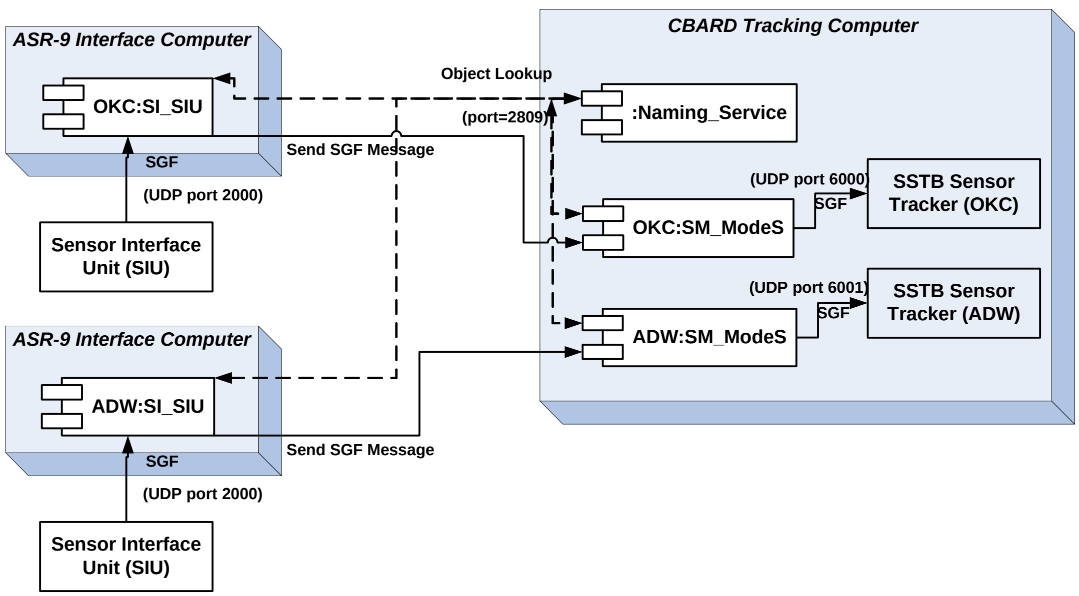

Figure 4-9. SDN software interface diagram.

Figure 4-10 illustrates SSTB software interfaces, including SM:SensorTracker, SensorTracker:Fusion Tracker, Fusion Tracker:SD, and FusionTracker:TDWRServer.

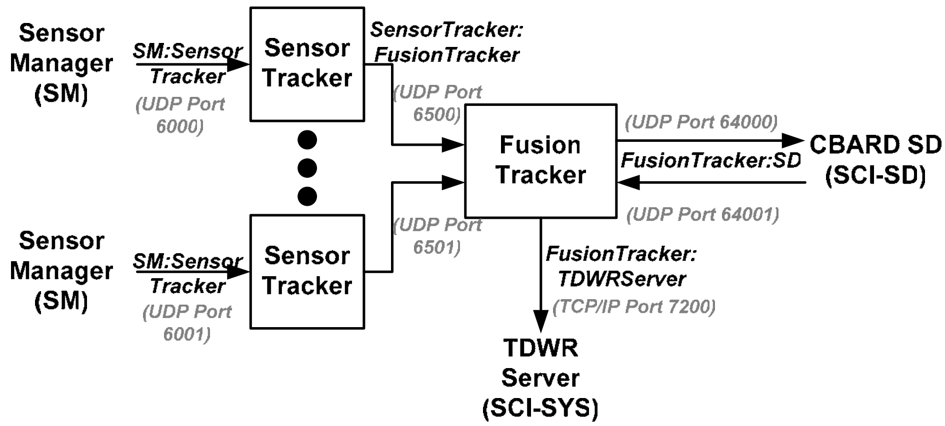

Figure 4-10. SSTB software interface diagram.

SIU:SI Interface

This paragraph describes the interface between the SENSIS Sensor Interface Unit (SIU) and the SDN Sensor Interface (SI) software. The SI software receives a real-time ASR-9 data stream by listening to a UDP/IP protocol broadcast. The example in Figure 4-9 illustrates the SIU:SI interface for two ASR-9 data streams. The UDP port address is configurable. For the prototype, port 2000 was selected. The SIU outputs the serial ASR-9 Common Digitizer (CD) message format [1] embedded within a 16-bit Surveillance Gateway Format (SGF) [2]. The SGF format puts a header around groups of CD messages.

SI:SM Interface

This paragraph describes the interface between the SDN SI and SM software components. The SI, executing on a computer at the TRACON, outputs the SGF format data with Common Object Request Broker Architecture (CORBA) wrappers, using the TCP/IP protocol over a wide-area Ethernet network. The SM, executing on a computer at the CBARD Command Center, listens to the appropriate TCP/IP port and receives SGF format ASR-9 aircraft surveillance data from the SI. The example in Figure 4-9 illustrates two SI:SM interfaces, one each for ASR-9 data from the OKC and ADW sites.

SI:NamingService Interface

The SI interfaces with the NamingServer in order to obtain the TCP/IP port information that makes it possible to communicate with the SM expecting data from a particular radar. The example in Figure 4-9 illustrates two SI:NamingServer interfaces, one each for ASR-9 data from the OKC and ADW sites. A wide-area Ethernet network provides the necessary communication path, since the SI runs at the TRACON and the NamingServer runs at the CBARD Command Center.

SM:NamingService Interface

The SM interfaces with the NamingServer in order to obtain the TCP/IP port information that makes it possible to communicate with the SI providing the ASR-9 data. The example in Figure 4-9 illustrates two SM:NamingServer interfaces, one each for ASR-9 data from the OKC and ADW sites. The SM and NamingServer run on computers at the CBARD Command Center. An Ethernet LAN is used to provide the communication path if the SM and NamingServer processes are not running on the same computer.

SM:SensorTracker Interface

This paragraph describes the interface between the SDN SM and SSTB Sensor Tracker software. This interface provides for the real-time data transfer of ASR-9 data from the SM to the Sensor Tracker software using a UDP broadcast protocol. As shown in Figure 4-9, each SM:SensorTracker interface carries data from a particular ASR-9 system. A different UDP/IP port address is used for each ASR-9 data stream. The SM and Sensor Tracker processes run on the same computer.

SensorTracker:FusionTracker Interface

A set of Sensor Tracker processes, one for each radar, interface with the Fusion Tracker in order to provide single-sensor track data streams for the Fusion Tracker to combine. The message format is described in Appendix A. As illustrated in Figure 4-10, each Sensor Tracker outputs its track data stream on a unique TCP/IP port, and the Fusion Tracker listens to all available ports. The Ethernet LAN at the CBARD Command Center provides the communication path.

FusionTracker:SD Interface

As illustrated in Figure 4-10, the SSTB Fusion Tracker software has a two-way interface with the CBARD Situational Display Software (SCI-SD). The Fusion Tracker sends aircraft track surveillance data messages to the SD via UDP/IP. The SD sends operator action messages on a different UDP/IP port to the Fusion Tracker that identify which track the CBARD Operator has selected using the SD CHI. The message formats are described in Appendix A. The Fusion Tracker and SD may execute on the same computer, or else on different computers connected via an Ethernet network.

FusionTracker:TDWRServer Interface

As illustrated in Figure 4-10, the SSTB Fusion Tracker software interfaces with TDWRServer software (SCI-SYS, System Control and Monitoring) in order to send cue messages indicating which track the CBARD Operator has selected using the SD CHI. The message format is described in Appendix A. The Fusion Tracker sends the cue message via TCP/IP using the SCLite inter-process communications library that is included as part of SCI-SHARE. The Fusion Tracker and TDWRServer software may execute on the same computer, or else on different computers connected via an Ethernet network.

5. TDWR DATA PROCESSING (SCI-TDWR)

5.1 OVERVIEW

This section describes the software design of the TDWR Data Processing Software (SCI-TDWR). This SCI contains the advanced algorithm technology necessary to command TDWR to scan for aircraft and then detect the aircraft and aerosol plume release signatures in the resulting TDWR Base Data imagery. Earlier, we referred to these algorithms as automated aircraft and plume detection. The algorithm that creates TDWR scan strategies for aircraft tracking is called the Three-Dimensional Aircraft Tracker (3-DAT). The algorithm that detects aircraft and plume signatures is called the Aerosol Release Detection Algorithm (ARDA). An overview of both algorithms is provided in the following paragraphs. The remainder is devoted to software architectural design.

5.1.1 Three-Dimensional Aircraft Tracker (3-DAT)

3-DAT is responsible for maintaining a three-dimensional scan strategy that allows TDWR to track an aircraft, and also sending this scan strategy to the TDWR RPG software. The algorithm has three input data sources. The first input is CBARD mode control commands from the Command Center that will enable or disable CBARD mode. This is used by the algorithm to start or stop processing. The second input is aircraft cues from the Command Center that will indicate the horizontal position of the threatening aircraft. This data is generated based on ASR-9 radar data. These messages will arrive approximately every 5 seconds (the ASR-9 antenna makes a single 360 degree rotation in approximately 4.65 seconds). The ASR-9 aircraft cue consists of a two-dimensional position, with altitude information for aircraft equipped with a beacon transponder. The third input is three-dimensional aircraft detections provided by ARDA. This input provides the best estimate of aircraft altitude, but is only available when a detection has been made from the TDWR data.

Aircraft tracking is performed using one of three tracking modes. The SEARCH mode is used to initially acquire the aircraft when CBARD mode is first enabled. The TRACK mode is used to provide tight tracking of the aircraft in order to support detection of potential plume signatures by ARDA. The TRACK mode is optimized to detect low flying aircraft at ranges greater than five kilometers. In the event that the aircraft flies closer than five kilometers to the radar, the algorithm will switch into a TRACK_CONE (short for cone of silence) mode that simplifies the scan selection process.

The algorithm will select the appropriate tracking mode based upon the availability of the aircraft detections from ARDA. The SEARCH mode is employed when the algorithm is enabled by the operator, or in the event that the track is lost on a previously tracked aircraft. A lost track is defined as three consecutive radar tilts where the aircraft signal was not extracted from the data. The TRACK mode is employed when the first aircraft detection is made by ARDA and is continually used until the track is lost, or the aircraft is closer than five kilometers to the radar. If the track is lost, the algorithm reverts to the SEARCH mode. If the aircraft is closer than five kilometers, the algorithm switches to the TRACK_CONE mode until the aircraft is lost or moves outside of five kilometers.

The SEARCH mode computes the elevation angle for several altitudes (defined in a parameter file) at the horizontal location of the aircraft. In the simplest scenario, the algorithm assumes that altitude information is not available in the ASR-9 based input, and scans these elevation angles repeatedly waiting for an aircraft detection from ARDA. However, if a beacon transponder based altitude is provided by the ASR-9, the algorithm will first select an elevation angle that matches the aircraft's reported altitude and horizontal position. The algorithm will also look 100 meters above and below the aircraft's reported altitude in order to accommodate a reasonable error in the reported altitude. If the aircraft is not located at any of these elevation angles, the algorithm will revert to the non-beacon case.

The TRACK mode is employed when the aircraft is detected by ARDA during the SEARCH mode and until the track is lost or CBARD mode is disabled. The TRACK mode consists of two elevation scans located near the aircraft's altitude as reported by ARDA. The first elevation angle is located one-half beam width (0.2 degrees) above the aircraft's altitude. The second elevation angle is located one-half beam width below the aircraft's altitude. Note that the algorithm does not scan directly at the aircraft's altitude.

In most cases, the strategy of looking slightly above and slightly below the aircraft will bracket the aircraft and provide solid consistent tracking. However, in some cases, the aircraft may move more than one-half beam width in the time it takes to perform the two scans. This often happens when the aircraft is moving quickly towards or away from the radar at higher altitudes. To prevent the loss of the aircraft track, the algorithm computes the elevation angle change during this time and if it is greater than half a beam width, the algorithm will modify the elevation angle in the appropriate direction to bracket the aircraft. The algorithm limits the two elevation scans to be within a beam width (0.5 degrees) of the last reported aircraft altitude.

The algorithm scans above and below the aircraft, rather than exactly at the detected aircraft altitude, for two reasons. Firstly, the goal is to maintain a track on the aircraft. Elevation angles with respect to the radar can change rapidly if the aircraft is in a climb or descent, or if its altitude is high relative to its range from the radar. Secondly, the primary goal of CBARD is to detect aerosol plumes behind the aircraft. Field trials have shown that the best plume signatures observed from the radar occur when the elevation angle is set to look slightly below the aircraft.

The TRACK_CONE mode is employed when the aircraft is closer than five kilometers to the TDWR. The TRACK_CONE mode consists of two elevation scans located at the aircraft's altitude. Due to the close proximity of the aircraft to the radar, any motion of the aircraft towards or away from the radar can cause rapid changes in elevation angle. The normal SEARCH mode will quickly lose the track as the limits of the elevation angles are reached. Therefore, SEARCH mode is only set up to maintain a track on an aircraft moving at level straight flight. The two elevation angles are placed at the forecasted position of the aircraft using the estimated altitude, speed and heading.

Once the algorithm chooses the elevation angles, it must then select the appropriate azimuths to start and stop the TDWR antenna. This is accomplished by creating a map of azimuths that the aircraft could potentially be located in during the scan strategy. First, the aircraft's current azimuth (relative to the TDWR) is added to the map. Then, the aircraft's forecasted position is computed by moving the aircraft forward at the current speed and 35 different headings rotating in 10 degree increments from 0 to 350 degrees. Finally, the location of a potential plume is determined. Nominally, it is assumed that a plume would last for 60 seconds. Once all of the azimuths are added to the map, the start and stop azimuth of the scan sector is simply read from the map. The final step is to check the size of the scan sector. If the sector is smaller than the minimum sector size (nominally 30 degrees), the sector is expanded to the minimum sector size. If the sector is larger than 180 degrees, the sector is expanded to a full volume scan of 360 degrees.

5.1.2 Aerosol Release Detection Algorithm (ARDA)

ARDA utilizes a knowledge-based engineering approach comprised of multi-dimensional image filtering, data fusion, delayed thresholding, and consensus building to identify aircraft and plume features in Doppler radar data. The key image filtering technique utilized in ARDA, Functional Template Correlation (FTC), was developed at Lincoln Laboratory and has been successfully used for a wide variety of automatic target recognition (ATR) applications and automated hazardous weather detection algorithms, including the Machine Intelligent Gust front Algorithm (MIGFA). MIGFA currently operates in several deployed FAA weather detection systems including TDWR, Integrated Terminal Weather System (ITWS), and the ASR-9 Weather System Processor (WSP).

We will present a brief overview of how the algorithm works, referring to the algorithm block diagram shown in Figure 5-1. ARDA employs image processing filters called "feature detectors" to detect features associated with the aircraft and plume in the input TDWR Base Data reflectivity and Doppler velocity imagery. The ASR-9 aircraft cue input is used to focus the search region within the Base Data in order to mitigate false detections. The algorithm employs several feature detectors, as shown in the figure. There are two groups of detectors: aircraft feature detectors and plume feature detectors. Each feature detector outputs an interest image, which is a map of probabilities that the particular feature is present at each location in the image. The interest images output by the aircraft detectors and plume detectors are separately combined using a weighted average in order to develop a consensus combined interest image. A thresholding operation is then performed prior to output of aircraft and plume detections.

Aircraft detection consists of looking for the bright reflectivity and high velocity region associated with the aircraft and tracking the region's scan-to-scan motion. The output of the aircraft detection algorithm is then used as input by the plume detection algorithm. Plume detection consists of looking for the weak line echo immediately behind the aircraft or in regions where a plume has been detected previously and may still be present. For example, the Initial Plume detector looks for the thin line signature in the initial release region immediately behind the aircraft in the current TDWR scan. The Frame-Integrated detector integrates the signal in a rectangular frame region following the aircraft in an attempt to bring out the plume signal with respect to the noise background, and then identifies the enhanced thin line of signal associated with the plume. To guard against false detections from pre-existing clutter or other atmospheric background structures (e.g. thermals) that may mimic the plume thin line signature, a Thin Line Suppressor feature detector (not shown in Figure 5-1) examines a rectangular region immediately ahead of the aircraft and creates an interest image of evidence in regions of pre-existing thin line features not associated with the aerosol release. If and when the aircraft subsequently flies through these regions, the disconfirming evidence will counteract any confirming evidence that may be generated from the other plume feature detectors.

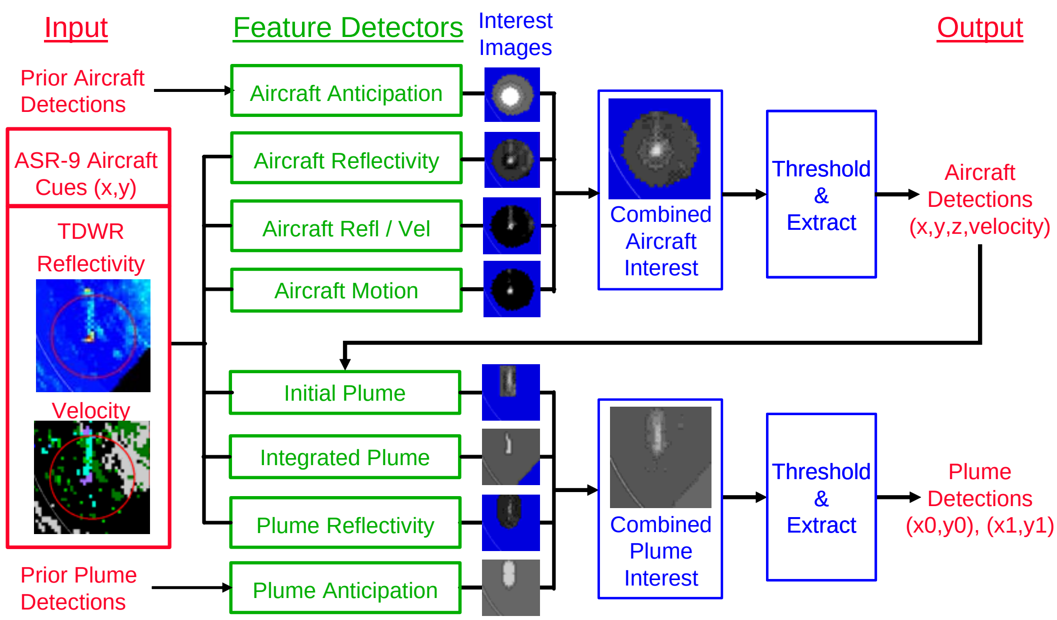

Figure 5-1. Aerosol Release Detection Algorithm (ARDA) block diagram.

ARDA also uses prior aircraft and plume detections to help anticipate future detections. Locally enhanced confirming evidence generated by the Aircraft Anticipation and Plume Anticipation feature detectors provides a means of selectively sensitizing the system in locations where an event is expected based on prior detections and allows detections to be made in areas where evidence might otherwise be too weak to survive thresholding. In addition to the localized boosting based on prior detections, the overall interest level of the anticipation images automatically fluctuates in response to internally computed background reflectivity statistics in order to regulate global detection sensitivity for optimal detection performance.

5.2 ARCHITECTURAL DESIGN

5.2.1 SCI-TDWR Components

SCI-TDWR consists of the following software components: ARDA, ARDAServer, 3-DAT, BDServer, and SCMuxMT. These components are discussed in the following paragraphs.

ARDA

ARDA performs the following functions within SCI-TDWR:

- Inputs CBARD mode control commands, ASR-9 aircraft location cues, and TDWR Base Data.
- Executes the aircraft and aerosol plume release detection algorithm (described above).
- Outputs aircraft and plume detection messages.

ARDA is a new algorithm developed at MIT Lincoln Laboratory for the CBARD project. An overview of the algorithm was provided earlier in this chapter.

ARDAServer

ARDAServer performs the following functions within SCI-TDWR:

- Inputs CBARD mode control commands, ASR-9 aircraft location cues, and TDWR Base Data, and forwards these data to ARDA.

ARDAServer is a message forwarding task that receives data from multiple input streams and then forwards the data on a single output stream to ARDA. Having a single data stream input to ARDA facilitates the archival of the data. Archived data can be used for playback and post-event analysis, with all data preserved in real-time order.

3-DAT

3-DAT performs the following functions within SCI-TDWR:

- Inputs ASR-9 aircraft location cues, TDWR aircraft detections, and CBARD mode control commands.
- Creates a three-dimensional scan strategy that allows the TDWR to track an aircraft.
- Outputs scan strategy commands to the TDWR RPG.

An overview of the algorithm was provided earlier in this chapter.

BDServer

BDServer performs the following functions within SCI-TDWR:

- Inputs TDWR Base Data from the TDWR RPG UDP broadcast socket.
- Outputs TDWR Base Data to ARDAServer via a TCP/IP socket.

BDServer is based on a similar translator developed for the Integrated Terminal Weather System (ITWS) program.

SCMuxMT

Two SCMuxMT tasks are run within SCI-TDWR. They perform the following functions:

– Forwards CBARD mode control commands and ASR-9 aircraft location cue messages from the CBARD Command Center System Control and Monitoring software (SCI-SYS) to the TDWR RPG.

– Forwards plume detection, TDWR aircraft location, and system status messages to CBARD Command Center System Control and Monitoring software (SCI-SYS).

The SCMuxMT tasks provide an interface between the CBARD Command Center and the TDWR by relaying messages. This software was originally developed for the ITWS program.

5.2.2 Interface Design

Interface Identification and Diagrams

Figure 5-2 illustrates the software interfaces for SCI-TDWR. The interface names appear along the data stream arrows in the figure. Interfaces with the external TDWR Radar Product Generator (RPG) software include TDWR_BD, RPG_CBMode, RPG_AircraftCue, TDWRCBOutput, TDWRCue, ScanStrategy, ARDA_AircraftDet, and ARDA_PlumeDet. Interfaces with the CBARD System Control and Monitoring Software (SCI-SYS) include TDWRCue:XXX and TDWRCBOutput:XXX. Internal interfaces within SCI-TDWR include TDWR_BD_SCAN, and ARDA_Server.

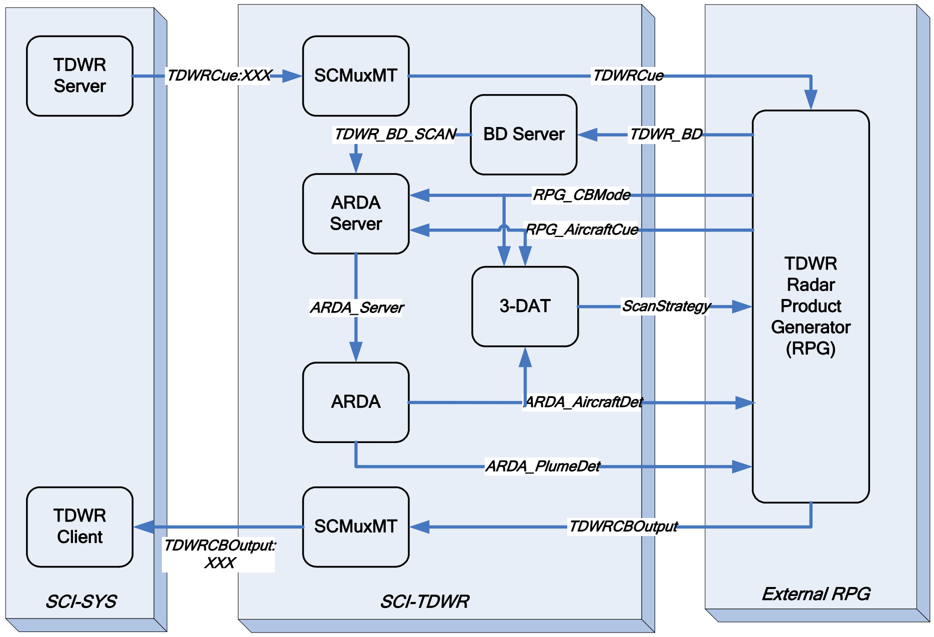

Figure 5-2. SCI-TDWR software interfaces.

It should be noted that SCI-TDWR as currently designed has a few software interfaces that could be eliminated with a modest amount of software development, resulting in a simpler design. In particular, the SCMuxMT message relay tasks were added late in the prototype development phase in order to implement a change in the CBARD network configuration to improve information security. The concern was that CBARD-specific data not be sent on the TDWR Base Data broadcast network, because the data could potentially be received by other non-FAA users of Base Data. All CBARD-specific messages were moved to the private TDWR Ethernet network that the RPG uses to communicate with the TDWR RMMS. Note that MIT Lincoln Laboratory and the FAA also plan to use the RMMS interface for the Radar Data Acquisition (RDA) upgrade. The message relay software tasks were the easiest way to implement this change in a way that minimized modifications to both CBARD and the RPG software.

TDWR_BD Interface

This paragraph describes the interface between the RPG and the BDServer software. This interface allows for the real-time transfer of TDWR Base Data from the RPG to CBARD. The RPG broadcasts TDWR Base Data on an Ethernet network using the UDP broadcast protocol. The BDServer software receives Base Data by attaching to the Base Data broadcast socket address. The format of TDWR Base Data is described in [8].

TABLE 5-1

TDWR_BD Interface Message Summary

Record Name	Comments	Format Reference
TDWR Base Data	TDWR Base Data.	Reference [8]

RPG_CBMode Interface

This paragraph describes the interface that the RPG uses to forward CBARD mode control commands (CBEnable, CBDisable) originated at the CBARD Command Center to ARDAServer and 3-DAT. Table 5-2 summarizes the messages and references which tables in Appendix A provide data formats. The interface is accomplished using the SCLite inter-process communication software library that is included as part of the Shared Software Library (SCI-SHARE) [9]. This SCLite interface uses the TCP/IP protocol over an Ethernet network. Note that the SCLite supports the concept of multiple recipients, and we consider this a single interface between one server and two clients.

TABLE 5-2

RPG_CBMode Interface Message Summary

Record Name	Comments	Recipients	Format Reference
CBEnable	Disable CBARD mode.	ARDAServer, 3-DAT	Table A-3
CBDisable	Enable CBARD mode.	ARDAServer, 3-DAT	Table A-4

RPG_AircraftCue Interface

This paragraph describes the interface that the RPG uses to forward ASR-9 aircraft location cue messages (CBAircraftLoc) originated at the CBARD Command Center to ARDAServer and 3-DAT. Table 5-3 summarizes the messages and references which tables in Appendix A provide data formats. The interface is accomplished using the SCLite inter-process communication software library (SCI-SHARE). This SCLite interface uses the TCP/IP protocol over an Ethernet network. Note that the SCLite supports the concept of multiple recipients, and we consider this a single interface between one server and two clients.

TABLE 5-3

RPG_AircraftCue Interface Message Summary

Record Name	Comments	Recipients	Format Reference
CBAircraftLoc	Location of aircraft as reported by the ASR-9 radar.	ARDAServer, 3-DAT	Table A-1

TDWRCBOutput Interface

This paragraph describes the interface between the RPG and SCMuxMT software. This interface allows for the real-time transfer of SCI-TDWR outputs, including plume detection messages (CBDetection), TDWR aircraft location messages (CBAircraftLoc), and system status messages (CBStatus) from the RPG to the SCMuxMT message relay software. SCMuxMT forwards these messages via the SCMuxMT:TDWRClient:Output interface. Table 5-4 summarizes the messages sent using this interface, and also references the tables in Appendix A that provide message format details. The interface is accomplished using the SCLite inter-process communication software library (SCI-SHARE). This SCLite interface uses the TCP/IP protocol over an Ethernet network.

TABLE 5-4

TDWRCBOutput Interface Message Summary

Record Name	Comments	Format Reference
CBAircraftLoc	Location of aircraft as reported by ARDA based on TDWR data.	Table A-1
FrameStart	Frame start message from ARDA aircraft or plume detection.	Table A-6
FrameEnd	Frame end message from ARDA aircraft or plume detection.	Table A-7
CBStatus	CBARD system status message generated by RPG based on TDWR system status and ARDA and 3-DAT SWStatus messages.	Table A-9
CBDetection	CB plume detection region and statistics.	Table A-2

TDWRCue Interface

This paragraph describes the interface between the SCMuxMT and TDWR RPG software. This interface allows for the real-time transfer of CBARD mode control commands (CBEnable, CBDisable) and ASR-9 aircraft location cue messages (CBAircraftLoc) from SCMuxMT to the RPG software. Table 5-5 summarizes the messages sent via this interface, and also references the tables in Appendix A that provide message format details. This interface is accomplished using the SCLite inter-process communication software library (SCI-SHARE). This SCLite interface uses the TCP/IP protocol over an Ethernet network. The SCMuxMT software is a message forwarding task, which receives the data via the TDWRCue:XXX interface.

TABLE 5-5

TDWRCue Interface Message Summary

Record Name	Comments	Format Reference
CBAircraftLoc	Location of aircraft as reported by the ASR-9 radar.	Table A-1
CBDisable	Disable CBARD mode.	Table A-3
CBEnable	Enable CBARD mode.	Table A-4

ScanStrategy Interface

This paragraph describes the interface between the 3-DAT and TDWR RPG software. This interface allows for the real-time transfer of scan strategy commands (CBScanStrategy) from 3-DAT to the RPG. Table 5-6 summarizes the messages sent via this interface, and also references the tables in Appendix A where message format details are provided. This interface is accomplished using the SCLite inter-process communication software library (SCI-SHARE). This SCLite interface uses the TCP/IP protocol over an Ethernet network.

TABLE 5-6

ScanStrategy Interface Message Summary

Record Name	Comments	Format Reference
CBScanStrategy	Scan strategy description to be forwarded by RPG to the TDWR DSP, antenna, and transmitter subsystems.	Table A-5
SWStatus	Software status message.	Table A-9

ARDA_ AircraftDet Interface

This paragraph describes the interface used by ARDA to output TDWR-based aircraft location data (CBAircraftLoc) to the TDWR RPG software and 3-DAT. Table 5-7 summarizes the messages sent via this interface, and also references the tables in Appendix A where message format details are provided. This interface is accomplished using the SCLite inter-process communication software library (SCI-SHARE). This SCLite interface uses the TCP/IP protocol over an Ethernet network. This is another SCLite interface that has a single server and two clients.

TABLE 5-7

ARDA_AircraftDet Interface Message Summary

Record Name	Comments	Recipients	Format Reference
FrameStart	Frame start messages indicates processing was performed. May be followed by CBAircraftLoc message.	RPG, 3-DAT	Table A-6
CBAircraftLoc	Location of aircraft as reported by ARDA based on TDWR data.	RPG, 3-DAT	Table A-1
FrameEnd	Frame end message.	RPG, 3-DAT	Table A-7
SWStatus	Software status message (i.e., heart beat) is sent every 30 seconds.	RPG, 3-DAT	Table A-8

ARDA_PlumeDet Interface

This paragraph describes the interface between the ARDA and TDWR RPG software that allows for the real-time transfer of TDWR plume detection messages (CBDetection) from the ARDA plume detection algorithms to the RPG. Table 5-8 summarizes the messages sent via this interface, and also references the tables in Appendix A where message format details are provided. This interface is accomplished using the SCLite inter-process communication software library (SCI-SHARE). This SCLite interface uses the TCP/IP protocol over an Ethernet network.

TABLE 5-8

ARDA_PlumeDet Interface Message Summary

Record Name	Comments	Format Reference
FrameStart	Frame start message. May be followed by CBDetection message.	Table A-5
CBDetection	CB plume detection region and statistics.	Table A-2
FrameEnd	Frame end message.	Table A-6

TDWRCue:XXX Interface

This paragraph describes the interface between the TDWRServer (SCI-SYS) and SCMuxMT software. This interface allows for the real-time transfer of CBARD mode control commands (CBEnable, CBDisable) and ASR-9 aircraft location cue messages (CBAircraftLoc) from the CBARD Command Center TDWRServer software to the SCI-TDWR SCMuxMT message relay software. The “XXX” in the interface name is a place holder for the 3-character site designation, since SCI-SYS maintains a separate interface with each CBARD-modified TDWR site. Table 5-9 summarizes the messages sent via this interface, and also references the tables in Appendix A where message format details are provided. This interface is accomplished using the SCLite inter-process communication software library (SCI-SHARE). This SCLite interface uses the TCP/IP protocol over an Ethernet network. The SCMuxMT software is a message forwarding task, which forwards the data to the RPG via the TDWRCue interface.

TABLE 5-9

TDWRCue:XXX Interface Message Summary

Record Name	Comments	Format Reference
CBAircraftLoc	Location of aircraft as reported by the ASR-9 radar.	Table A-1
CBDisable	Disable CBARD mode.	Table A-3
CBEnable	Enable CBARD mode.	Table A-4

TDWRCBOutput:XXX Interface

This paragraph describes the interface between the SCMuxMT message relay software and the CBARD Command Center TDWRClient (SCI-SYS) software. This interface allows for the real-time transfer of SCI-TDWR outputs, including plume detection messages (CBDetection), TDWR aircraft location messages (CBAircraftLoc), and system status messages (CBStatus) to the CBARD Command Center. SCMuxMT is a message forwarding task, which receives the data via the RPG:SCMutMT:Output interface. The “XXX” in the interface name is a place holder for the 3-character site designation, since SCI-SYS maintains a separate interface with each CBARD-modified TDWR site. Table 5-10 summarizes the messages sent via this interface, and also references the tables in Appendix A where message format details are provided. The interface is accomplished using the SCLite inter-process communication software library (SCI-SHARE). This SCLite interface uses the TCP/IP protocol over an Ethernet wide-area network.

TABLE 5-10

TDWRCBOutput:XXX Interface Message Summary

Record Name	Comments	Format Reference
FrameStart	Frame start message from ARDA aircraft or plume detection. This message may be followed by a CBAircraftLoc message.	Table A-6
CBAircraftLoc	Location of aircraft as reported by ARDA based on TDWR data.	Table A-1
FrameEnd	Frame end message from ARDA aircraft or plume detection.	Table A-7
CBStatus	CBARD system status message generated by RPG based on TDWR system status and ARDA and 3-DAT SWStatus messages.	Table A-9
CBDetection	CB plume detection region and statistics.	Table A-2

TDWR_BD_SCAN Interface

This paragraph describes the interface between the BDServer and ARDAServer software. This interface allows for the real-time transfer of reformatted TDWR Base Data to the SCI-TDWR CBARD algorithms. Table 5-11 summarizes the messages sent via this interface, and references where message format details are provided. The interface is accomplished using the SCLite inter-process communication software library (SCI-SHARE). This SCLite interface uses the TCP/IP protocol over an Ethernet local-area network.

TABLE 5-11

TDWR_BD_SCAN Interface Message Summary

Record Name	Comments	Format Reference
TDWR Base Data	TDWR Base Data.	Reference [8]

ARDA_Server Interface

This paragraph describes the interface between the ARDAServer and ARDA software. This interface allows for the real-time transfer of TDWR Base Data, CBARD mode control commands (CBEnable, CBDisable), and ASR-9 aircraft location cue messages (CBAircraftLoc) to the aircraft detection and plume detection algorithms. Table 5-12 describes the messages that are sent via this interface, and also references the tables in Appendix A where message format details are provided. The interface is accomplished using the SCLite inter-process communication software library (SCI-SHARE). This SCLite interface uses the TCP/IP protocol over an Ethernet local-area network.

TABLE 5-12

ARDA_Server Interface Message Summary

Record Name	Comments	Format Reference
CBAircraftLoc	Location of aircraft as reported by the ASR-9 radar.	Table A-1
CBDisable	Disable CBARD mode.	Table A-3
CBEnable	Enable CBARD mode.	Table A-4
TDWR Base Data	TDWR Base Data.	Reference [8]

5.3 FAA TDWR SOFTWARE MODIFICATION

The TDWR Radar Product Generator (RPG) software subsystem, which runs on an SGI Origin 200 computer, is a collection of software tasks that perform base data generation and meteorological data processing. The FAA developed a modification to the RPG software in order to add the interfaces and functionality required for TDWR CBARD operations. The main functions of the RPG for CBARD are:

- Receiving CBARD mode control commands and ASR-9 aircraft cue messages via the SCMuxMT:RPG:Cue software interface.
- Activating CBARD mode in response to a CBARD mode control command, and deactivating CBARD mode in response to a mode control command or in the event of a TDWR system error.
- Forwarding ASR-9 aircraft cue messages to the SCI-TDWR ARDA and 3-DAT software tasks via the RPG:ARDAServer and RPG:3-DAT software interfaces.
- Executing CBARD scan strategy commands received via the 3-DAT:RPG interface.
- Receiving plume detection, TDWR aircraft detection, and software status messages from SCI-TDWR via the ARDA:RPG:Plume and ARDA:RPG:Aircraft software interfaces.
- Creation of CBARD status messages.
- Output of plume detection, aircraft detection, and CBARD status messages via the RPG:SCMuxMT:Output software interface.

6. CBARD SITUATIONAL DISPLAY (SCI-SD)

6.1 OVERVIEW

This section describes the software design for the CBARD prototype implementation of SCI-SD, the CBARD Situational Display (SD) Software. The CBARD SD software is built upon the Raytheon-Solipsys Tactical Display Framework (TDF) software development kit. TDF is a Java based user interface development package that provides a powerful set of built-in java classes and a framework that allows developers to extend the display capabilities. Note that the prototype system described in this document uses TDF version 3.7, while the current version is 4.x. If CBARD were to become a production system, the display software should be updated to use the current version of TDF.

The CBARD SD allows for the display of fused aircraft tracks from ASR-9 radars for a particular site on a plan display. The individual aircraft tracks are selectable via the computer mouse to obtain additional information about a particular track. While there may be multiple sites implemented in the CBARD network, the initial version of the system only allows one site to be actively engaged in CBARD surveillance at any given time. The CBARD surveillance function is enabled by selecting a track on the display and then pressing a single button. During operation and during tracking of aircraft the display shows current site status and current tracking status to the user via a status panel and colors on the display. A snapshot of the CBARD SD is shown in Figure 6-1.

The rest of this chapter describes the SD software architectural design.

6.2 DESIGN DECISIONS

The key factor in determining the CBARD SD design was the need to develop, test, and deploy quickly. It was deemed beneficial that the display have a user interface that was familiar to users of other similar applications. The SD must provide an easy to use interface that facilitates rapid situational awareness and a very straightforward procedure to allow the operator to select a track and cue the TDWR radar under stressful conditions. TDF satisfied all of these conditions. Software developers at MIT Lincoln Laboratory had recent experience with TDF on other applications.

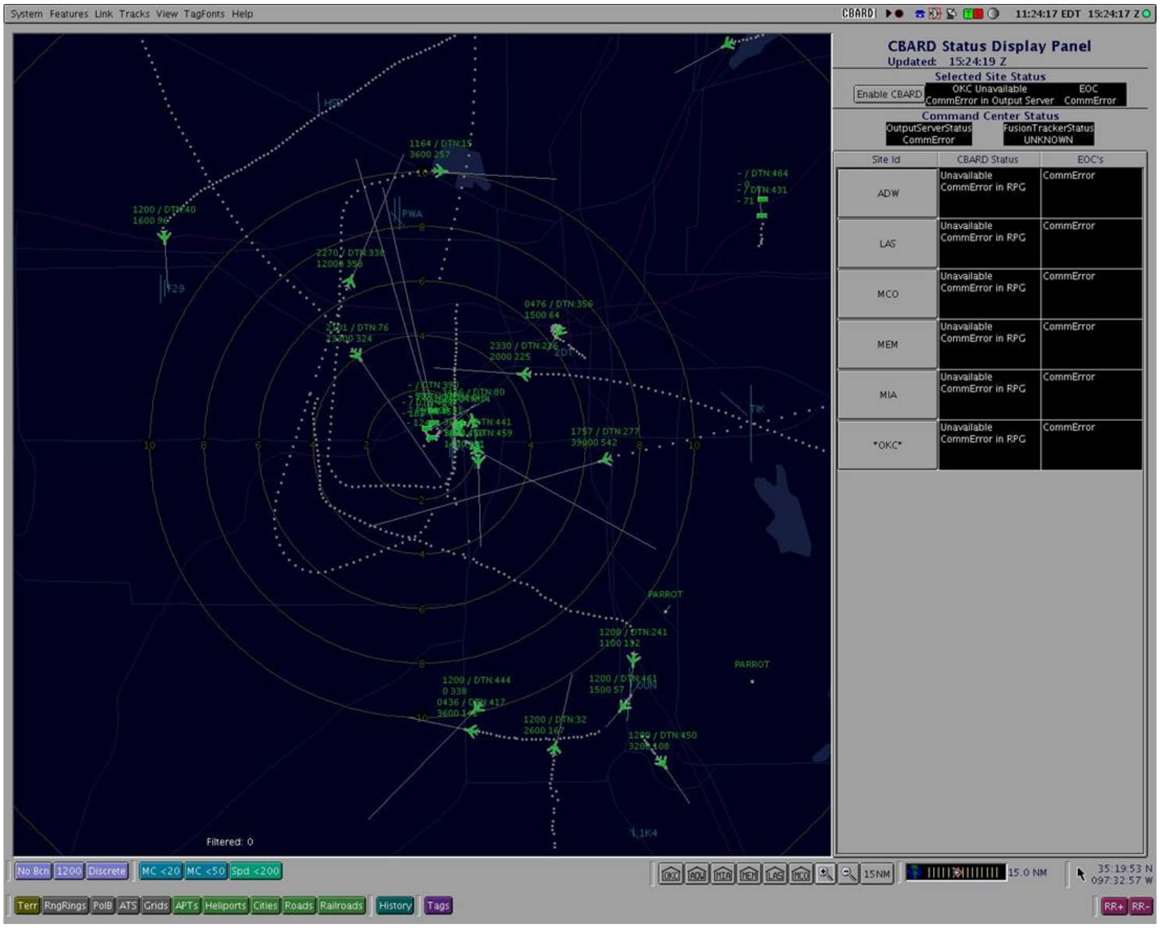

Figure 6-1. The CBARD situational display.

6.3 ARCHITECTURAL DESIGN

The architecture of the display is based upon the "model – view" paradigm. Relevant status and detection data are received on one thread of execution, which then updates a data model with the new status. Upon update, the data model then issues notifications to all registered listeners, in this case the CBARD status console, which displays the information to the user on another thread of execution. Each SD component will now be described.

6.3.1 SCI-SD Components

The CBARD situation display is a multi-threaded java application. The Graphical User Interface (GUI) contains 'off the shelf' modules from TDF, and custom modules developed for CBARD. There are three main modules that were developed for CBARD: the status ingest; the status buffer model; and the display console. The status ingest reads status information off a TCP/IP stream. The status buffer model stores current status information for each site. The display console (the GUI) is a java panel. Other elements of the display which are used from the TDF library are described below. Other customizations include new icons for aircraft and airport locations, the display of airport runways at close ranges, and the addition of range rings around the CBARD site. The SD components and interfaces are illustrated in Figure 6-2.

The CBARD display is broken up into four main sections observable by the user: the pull down menu bar at the top of the display; the TACSIT (tactical situation); the CBARD console; and the bottom button bar. The TACSIT shows the plan view of aircraft tracks, map features, airports, other surface features and overlays such as range rings and roads. The CBARD console shows the status of operational sites, allows for the switching of the active site, and allows the activation of the CBARD tracking system for specific aircraft threats. The bottom button bar allows for the filtering of aircraft tracks based on speed and altitude, the display of history trails showing previous locations of aircraft, and other map features such as terrain.

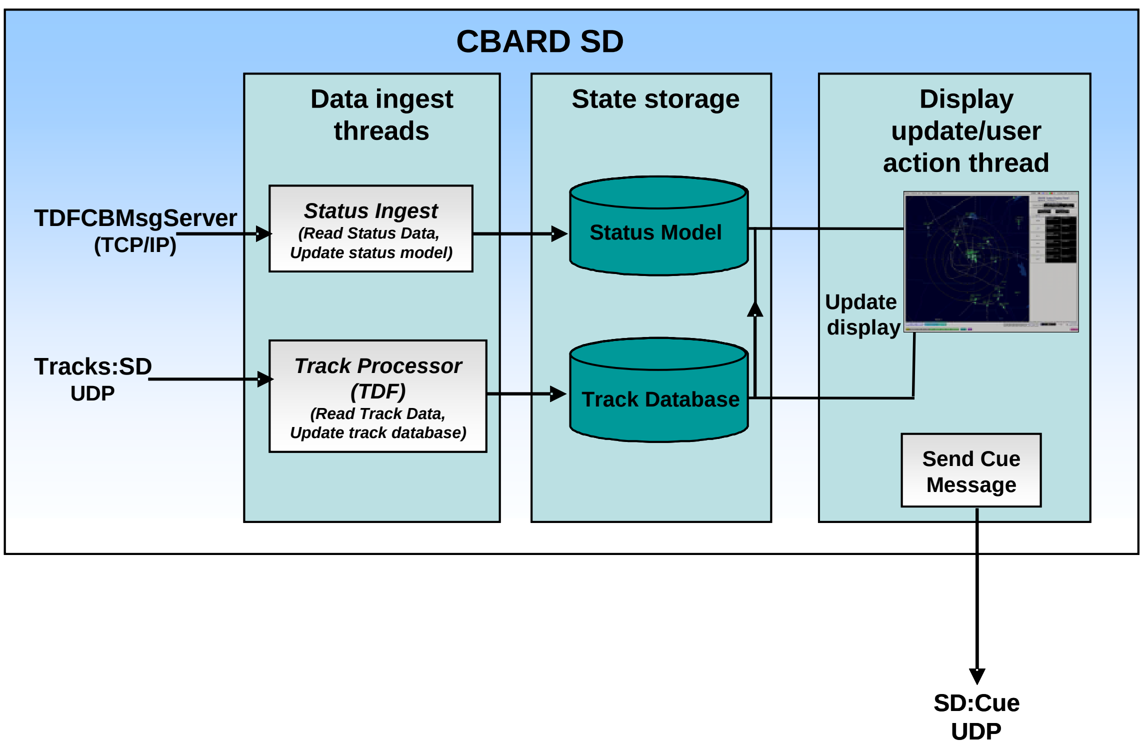

Figure 6-2. CBARD SD software components and interfaces.

Status Ingest

This module runs as a separate thread and continually reads status messages sent via TCP/IP from the TDFCBMsgServer interface. All CBARD subsystems report status to the display using the CBStatus message format for both user awareness and system diagnostics. Other messages which are received and processed are Aircraft Location messages (CBAircraftLoc) that are produced when the TDWR is tracking an aircraft, and Aerosol Release Detection (CBDetection) messages that are produced when a plume detection is reported from the TDWR. These messages are processed and the data model is updated. The display panel is then updated with the new information. Message format details are provided in Appendix A.

The only configuration option for the status ingest is the hostname and port number of the TDFCBMsgServer computer. This information is read from the configuration file *CBARD.prp*.

As previously mentioned, CBStatus messages are received via the TDFCBMsgServer interface for all modules in the CBARD system. Messages arrive on the TCP stream from each operational TDWR, and from the NBCMsgServer (part of SCI-OUT). The NBCMsgServer sends two types of CBStatus messages: a ‘heartbeat’ which is sent on a 30 second strobe to report on the status and connectivity of the process itself; and a report on the status of communications to the EOC computers. In the case of the ‘heartbeat’ function the ‘radarName’ field in the message is set to ‘NBCMsgServer’. When detection messages are sent to the EOC computer the NBCMsgServer will send back to the display a status message with the ‘radarName’ field set to the EOC name. This message will report on the success of the communication to a particular EOC computer.

CBARD Status Model

The status model stores current information for all active CBARD sites which includes current status as well as static information such as the site location and site name. The site information is read from the file *sites.xml* that is specified in the *CBARD.prp* properties file.

CBARD Display Console

This custom panel is a ‘listener’ to the CBARD status model that receives updates about the current state of the system. This panel also contains buttons which allow the user to enable and disable the system and switch active sites. The text on the button changes depending on the “enable” status of the CBARD system. The status console is shown as it looks at startup in Figure 6-3. Once the display starts receiving status messages the status of each module is displayed in the appropriate location on the console. The top of the console shows the title and the last time the console was updated. The next section shows information for the current selected site. From left to right are the enable button, the site status, and the Emergency Operation Center (EOC) status. The enable button is used to enable the system to track and monitor an aircraft after the aircraft is selected on the main TACSIT display. The next section below the site status is the command center status which displays the status of the output server and fusion tracker status. Note that in the prototype version of CBARD the fusion tracker status was not yet available. Below the command center status section is a section for each operational site. For each site there is a button to switch to that site, the site's status and the status of the corresponding EOC. Clicking on the site's button will change the display to that site, which will re-center the TACSIT and show the status for that site at the top of the panel. The site buttons are disabled when CBARD tracking is enabled. The CBARD function must be disabled before changing the active site.

The site information is read by reading the file *sites.xml*. The location of this file is specified in the properties file *CBARD.prp*.

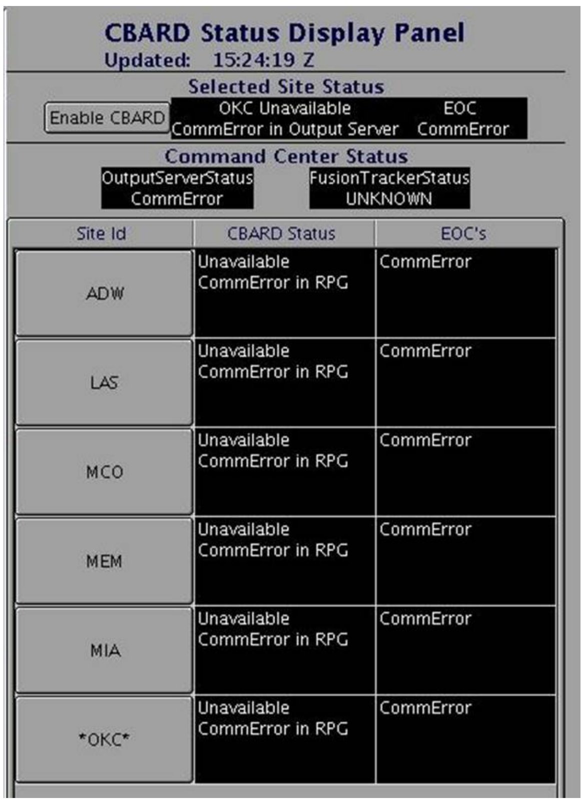

Figure 6-3. CBARD site status console at startup showing all site unavailable.

The following paragraphs describe some of the main built-in TDF display features, and some of the customizations developed for CBARD. The top menubar allows for system settings. The only one which operational users of the display would use is the 'TagFonts' menu that allows for the adjustment of the size of the text used for the aircraft tags on the TACSIT display.

The tactical situation display (TACSIT) shows the current location of aircraft targets on a plan view with other geographical information. Virtually all aspects of the TACSIT can be configured by the two XML files *config.xml* and *view.xml*. This section describes the more important configurable elements of the display. The TACSIT and related buttons at the bottom of the TACSIT are TDF functions that have been customized for use with CBARD. Customization of the display is done through XML configuration files, which are read at run-time to add and configure functionality to the display.

The map toolbar is shown in detail in Figure 6-4. This toolbar is configured in the configuration file *config.xml*. The 'range ring' and 'airport' buttons access customized code for CBARD; the remaining buttons are from TDF. The buttons are described from their abbreviations in Table 6-1. The configuration of this toolbar is set in the *config.xml* file. The range ring code and markers for the TDWR radars (shown at the center of the range ring) are configured in the *view.xml* file.

Figure 6-4. Map toolbar at bottom of TACSIT.

TABLE 6-1

Map Toolbar Button Descriptions

Button	Description
Terr	Terrain
RngRings	Range Rings
PolB	Political Boundaries
ATS	Air Traffic Routes
Grids	Latitude/Longitude Grid Lines
APTs	Airport Locations Note: when TacSit less than 15nm range runways are displayed.
Heliports	Heliport Locations
Cities	Major Cities
Roads	Roads
Railroads	Railroads
History	Track history trails (2 minutes)
Tags	Tag text for aircraft tracks

The filter toolbar is shown in detail in Figure 6-5, with descriptions of the buttons in Table 6-2. The configuration of this toolbar is set in the *config.xml* file.

Figure 6-5. Filter toolbar at bottom of TACSIT.

TABLE 6-2

Filter Toolbar Button Descriptions

Button	Description
No Bcn	Search-only (primary radar) targets
1200	Show only aircraft with 1200 beacon identity codes
Discrete	Show only aircraft with Discrete beacon codes
MC < 20	Show only aircraft with Mode C altitude less than 2000 ft.
MC < 50	Show only aircraft with Mode C altitude less than 5000 ft.
Spd < 200	Show only aircraft with speed less than 200 knots

The site and location toolbar is shown in detail in Figure 6-6, with descriptions of the buttons in Table 6-3. The configuration of this toolbar is set in the *config.xml* file.

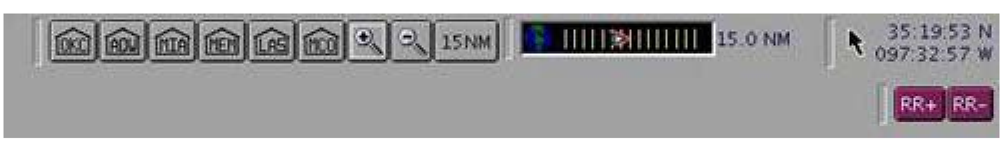

Figure 6-6. Site and location toolbar detail.

Some additional customizations done within the *config.xml* file include the addition of a custom renderer of aircraft symbols (symbol factory), configuration of the data displayed in the aircraft tag shown on the TACSIT, configuration of the data displayed in the selection detail window shown when aircraft are selected, and the configuration of the data displayed when the computer mouse is 'rolled over' an aircraft in the TACSIT.

TABLE 6-3

Location Toolbar Button Descriptions

Button	Description
OKC, …, MCO	Change active site. Note: not enabled while CBARD is enabled for one site. Note: These buttons were planned to be removed from this toolbar, because they are also available on the CBARD Status panel.
Magnifier Glass (+/-)	Zoom TacSit display in and out.
15NM	Quick range change to 15 nm.
Eye control	Click and drag to zoom in/out.
Lat/Lon	Shows latitude/longitude location of cursor in TacSit.
RR+ / RR-	Change range of 'track selection circle'.

6.3.2 Concept of Execution

The concept of execution of the display is such that after receiving direction of the location of a suspect aircraft, the operator selects the relevant site of interest from the status console on the right hand side of the display, and the aircraft of interest is then selected ('hooked') on the TACSIT. The selected aircraft is then highlighted with a white circle surrounding the track, as shown in Figure 6-7. When it is determined that this aircraft should be tracked by the TDWR and the CBARD ability should be enabled, the 'Enable CBARD' button is pressed. This will start the tracking function of the TDWR. When the TDWR has switched into CBARD tracking mode, the circle around the aircraft will turn green. Once the TDWR locks on the aircraft the circle around the aircraft will turn yellow. If ARDA reports a detection from this aircraft, the circle will turn red.

Figure 6-7. CBARD display showing a selected track.

6.3.3 Interface Design

Interface Identification and Diagrams

Figure 6-8 (below) and figure 6-2 (above) illustrate the software interfaces for SCI-SD.

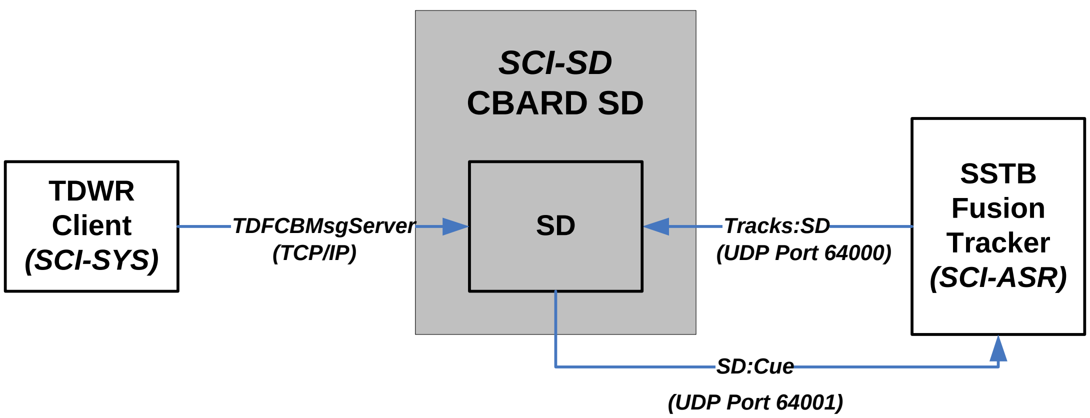

Figure 6-8. SCI-SD software interfaces.

Tracks:SD Interface

The Tracks:SD interface allows the CBARD SD to receive aircraft track data from the SCI-ASR SSTB Fusion Tracker software. The CBARD SD receives track data on a UDP stream and processes it with the built-in TDF track processor. These tracks are stored in a track database within TDF for access from other functions such as the Cue (enable CBARD) function.

TDFCBMsgServer Interface

This paragraph describes the interface between the SCI-SYS TDWRClient software and the SD. The SD receives System Status (CBStatus), Aircraft Location (CBAircraftLoc), and Aerosol Release Detection (CBDetection) messages from SCI-SYS. This interface is also described in section 7.

SD:Cue Interface

This paragraph describes the interface between the SD and the SCI-ASR SSTB Track Fusion software responsible for receiving data on CBARD operator actions. Cue messages are sent from the CBARD display to SCI-ASR via a message *CBARDCueRadarMessage*, which is an extension of the TDF *DisplayForceTellMessage*. This extended class adds an (4-byte) integer site index (corresponding to the site index value in the *sites.xml* configuration file), and the (16-byte character array) site name. The other relevant data is the track number (a 2-byte integer) of the aircraft for the TDWR to track. The addition of the site id was need for the multiple site feature added to CBARD. This message is sent to SCI-ASR via the post office function within TDF. The message is passed to the default post office which sends it on the same UDP port over which the track data are received.

7. SYSTEM CONTROL AND MONITORING (SCI-SYS)

7.1 OVERVIEW

This section describes the software design for the CBARD System Control and Monitoring Software (SCI-SYS). The primary functions of SCI-SYS are:

- To provide a mechanism to send CBARD control commands and data to each TDWR connected to the CBARD network.
- To facilitate the monitoring of CBARD hardware and software component status throughout the CBARD network from the CBARD Command Center facility.

7.2 ARCHITECTURAL DESIGN

7.2.1 SCI-SYS Components

SCI-SYS consists of three software components: TDWRServer, TDWRClient, and MonitorRecv. The TDWRServer software performs the CBARD control function. The TDWRClient software facilitates the monitoring function. The third component, MonitorRecv, was created for use as a remote monitoring aid during the CBARD prototype phase, when MIT Lincoln Laboratory personnel would be responsible for system maintenance.

TDWRServer

TDWRServer performs the following functions within SCI-SYS running on the CBARD SD computer (HWCI-SD):

- Maintains communication connectivity to every TDWR connected to the CBARD network via software interfaces with the SCI-TDWR software running at each TDWR facility.
- Receives CBARD mode control commands and ASR-9 aircraft location cues from SCI-ASR (originated by the CBARD Operator via the SD).
- Forwards messages received from SCI-ASR to the appropriate SCI-TDWR to allow CBARD mode to be executed at that TDWR facility.

TDWRClient

TDWRClient performs the following functions within SCI-SYS running on HWCI-SD:

- Maintains communication connectivity to every TDWR connected to the CBARD network via software interfaces with the SCI-TDWR software running at each TDWR facility.

- Receives CBARD status messages, TDWR-based aircraft detection messages, and plume detection messages from the SCI-TDWR software interfaces.

- Forwards messages received from each SCI-TDWR interface to SCI-SD and SCI-SYS, as appropriate.

MonitorRecv

MonitorRecv performs the following functions within SCI-SYS running on the Remote Monitoring computer (HWCI-RMON):

- Receives CBARD status messages, TDWR-based aircraft detection messages, and plume detection messages from the SCI-OUT CBOutputRecv software via the CBARD User Network.

- Provides a display of CBARD system status.

- Logs received messages to a hard disk.

7.2.2 Interface Design

Interface Identification and Diagrams

Figure 7-1 illustrates the software interfaces for SCI-SYS. The interface names appear in the diagram along the connecting lines between software units. The TDWRServer software has two software interfaces. The TDWRCue:XXX interface is used to send CBARD mode control commands and aircraft location cues to any TDWR equipped for CBARD operations. The CBCue interface allows TDWRServer to acquire the control commands and aircraft location cues from SCI-ASR. The TDWRClient software has three software interfaces. The TDWRCBOutput:XXX interface allows plume detections and status information to travel from a TDWR site the CBARD Command Center. The TDFCBMsgServer interface allows this same data to reach SCI-SD. The OutputServer interface allows a two way exchange of data between the SD computer (HWCI-SD) and the output server computer (HWCI-OUT). The MonitorRecv software has a single software interface, the Remote Monitor interface, with SCI-OUT.

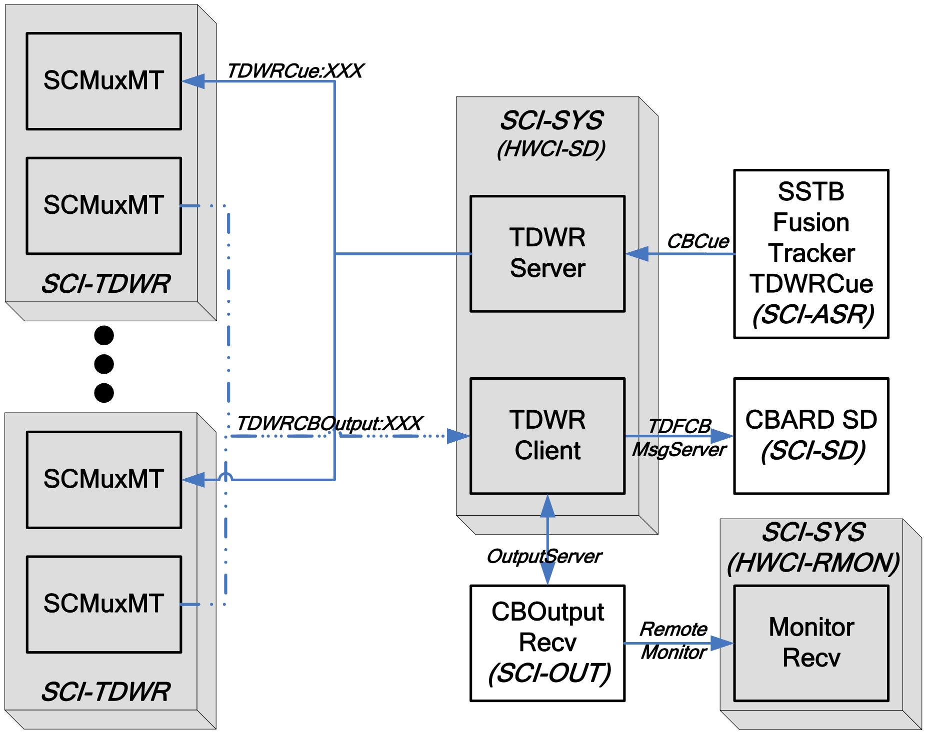

Figure 7-1. SCI-SYS software interfaces.

TDWRCue:XXX Interface

This paragraph describes the interface between the TDWR Server software (SCI-SYS) and the SCI-TDWR SCMuxMT software. This interface is used by TDWRServer to send CBARD mode control commands (CBEnable, CBDisable), and ASR-9 based aircraft location cue (CBAircraftLoc) messages to SCI-TDWR at each CBARD-equipped TDWR facility. The "XXX" in the interface name is a place holder for the 3-character site designation of each TDWR site. Note that this interface was also described in the chapter on TDWR Processing. For convenience, Table 7-1 summarizes the messages sent via this interface, and also references the tables in Appendix A where message format details are provided. The TDWRCue:XXX interface is implemented using the SCLite inter-process communication software that is included in SCI-SHARE. This particular SCLite communication channel uses the TCP/IP protocol over the private Ethernet WAN connecting the CBARD Command Center with the TDWR facility.

TABLE 7-1

TDWRCue:XXX Interface Message Summary

Record Name	Comments	Format Reference
CBAircraftLoc	Location of aircraft as reported by the ASR-9 radar.	Table A-1
CBDisable	Disable CBARD mode.	Table A-3
CBEnable	Enable CBARD mode.	Table A-4

TDWRCBOutput:XXX Interface

This paragraph describes the interface between the SCMuxMT (SCI-TDWR) and TDWRClient (SCI-SYS). This real-time data transfer interface allows the TDWR Processing software to output plume detection (CBDetection), TDWR-based aircraft location (CBAircraftLoc), and CBARD system status (CBStatus) messages to SCI-SYS to facilitate system monitoring. Note that this interface is also described in the chapter on TDWR Processing. For convenience, Table 7-2 summarizes the messages sent via this interface, and also references the tables in Appendix A where message format details are presented. The "XXX" in the interface name is a place holder for the 3-character site designation, since TDWRClient maintains separate interfaces with the SCI-TDWR software running at each CBARD-equipped TDWR facility. Figure 7-1 above illustrates this relationship. The TDWRCBOutput:XXX interface is implemented using the SCLite inter-process communication software that is included in SCI-SHARE. The TCP/IP protocol is used on the private Ethernet WAN connecting the CBARD Command Center with the TDWR facility.

TABLE 7-2

TDWRCBOutput:XXX Interface Message Summary

Record Name	Comments	Format Reference
CBAircraftLoc	Location of aircraft as reported by ARDA based on TDWR data.	Table A-1
FrameStart	Frame start message from ARDA aircraft or plume detection.	Table A-6
FrameEnd	Frame end message from ARDA aircraft or plume detection.	Table A-7
CBStatus	CBARD system status messages generated by RPG based on TDWR system status and ARDA and 3-DAT SWStatus messages.	Table A-9
CBDetection	CB plume detection region and statistics.	Table A-2

CBCue Interface

This paragraph describes the interface between the SCI-ASR SSTB TDWRCue task. This interface allows the ASR-9 Processing Software (SCI-ASR) to send CBARD mode control commands (CBEnable, CBDisable) and ASR-9 based aircraft location cue messages (CBAircraftLoc) to TDWRServer. TDWRServer will then forward these messages to the appropriate TDWR site via the TDWRCue:XXX interface described above. Table 7-3 summarizes the messages that are sent via this interface, and also references the tables in Appendix A that provide message format details. This interface uses the SCLite software and the TCP/IP protocol via an Ethernet LAN at the CBARD Command Center.

TABLE 7-3

CBCue Interface Message Summary

Record Name	Comments	Format Reference
CBAircraftLoc	Location of aircraft as reported by the ASR-9 radar.	Table A-1
CBDisable	Disable CBARD mode.	Table A-3
CBEnable	Enable CBARD mode.	Table A-4

TDFCBMsgServer Interface

This paragraph describes the interface between the TDWRClient and SCI-SD. This interface allows TDWRClient to forward system status (CBStatus), plume detection (CBDetection), and TDWR-based aircraft location (CBAircraftLoc) messages to SCI-SD. The “TDF” in the interface name refers to the Tactical Display Framework software upon which the SD is built. Table 7-4 describes the messages sent via this interface, and also references the tables in Appendix A where message format details are provided. This interface uses SCLite with the TCP/IP protocol. The TDWRClient is the server side of an SCLite TCP/IP stream, and the SD is the client.

Note that there are four varieties of CBStatus messages sent via the TDFCBMsgServer interface to the SD, as follows:

1. Status messages originated by the TDWR site that indicate the state of the TDWR and associated CBARD TDWR Processing Software (SCI-TDWR).

2. Status messages originated by the CBOutputRecv (SCI-OUT) software once every 30 seconds that indicate that the OutputServer interface (described below) is functioning properly.

3. Status messages originated by the NBCMsgServer (SCI-OUT) software once every 30 seconds that indicate that NBCMsgServer is functioning properly.

4. Status messages originated by the NBCMsgServer (SCI-OUT) software after each attempt to transmit a plume detection or status message to the NBCAnalysis software located at an end-user EOC facility.

TABLE 7-4

TDFCBMsgServer Interface Message Summary

Record Name	Comments	Format Reference
CBAircraftLoc	Location of aircraft as reported by ARDA based on TDWR data.	Table A-1
CBStatus	CBARD system status messages generated by RPG, CBOutputRecv (SCI-OUT), or NBCMsgServer (SCI-OUT).	Table A-9
CBDetection	CB plume detection region and statistics.	Table A-2

OutputServer Interface

This paragraph describes the two-way interface between TDWRClient and SCI-OUT. TDWRClient sends CBARD system status (CBStatus) and plume detection (CBDetection) messages to SCI-OUT. TDWRClient receives system status (CBStatus) messages from SCI-OUT. Referring to the four system status message types described in the previous paragraph, TDWRClient sends type 1 to SCI-OUT, and receives types 2, 3, and 4 from SCI-OUT. Table 7-5 summarizes the messages that are sent via this interface, and also references the tables in Appendix A where message format details are provided. This interface utilizes a two-way serial RS-232 cable connection between the Situational Display computer (HWCI-SD) and the Output Dissemination computer (HWCI-OUT). A set of TTY utility subroutines are used on both sides of the interface to perform open, close, read, and write functions over a serial link.

Plume Detection messages are only output if an aerosol release is detected at a TDWR facility. For a single CBARD Tracking Mode mission with a selected aircraft for a single TDWR facility, only a single Plume Detection message can be generated. System Status messages are forwarded from SCI-SYS to SCI-OUT approximately every 30 seconds for each TDWR site, and also anytime the CBARD Operator enables or disables the CBARD Tracking Mode at a particular site.

SCI-OUT sends three distinct subtypes of System Status (CBStatus) messages to SCI-SYS. A status message reporting the status of the CBOutputRecv software task that executes as part of SCI-OUT arrives once every 30 seconds. A status message reporting the status of the NBCMsgServer software task that executive as part of SCI-OUT also arrives at 30 second intervals. Finally, a status message reporting the status of a particular EOC NBCAnalysis external system is generated every time an attempt is made to transmit a message to one of them.

TABLE 7-5

OutputServer Interface Message Summary

Record Name	Comments	Format Reference
CBStatus	CBARD system status messages generated by RPG are sent from SCI-SYS to SCI-OUT. System status messages generated by CBOutputRecv or NBCMsgServer are sent from SCI-OUT to SCI-SYS.	Table A-9
CBDetection	CB plume detection region and statistics.	Table A-2

RemoteMonitor Interface

This paragraph describes the interface between the SCI-OUT CBOutputRecv software and the SCI-SYS MonitorRecv software. SCI-OUT forwards the status (CBStatus) and plume detection (CBDetection) messages it has received from the SCI-SYS TDWRClient software to the remote monitoring component of SCI-SYS running on HWCI-RMON. Table 7-6 summarizes the messages that are sent via this interface, and also references the tables in Appendix A where message format details are provided. This interface utilizes the SCI-SHARE SCLite inter-process communications library to implement a TCP/IP link along the CBARD User Network.

TABLE 7-6

RemoteMonitor Interface Message Summary

Record Name	Comments	Format Reference
CBStatus	CBARD system status messages.	Table A-9
CBDetection	CB plume detection region and statistics.	Table A-2

8. OUTPUT DISSEMINATION (SCI-OUT)

8.1 OVERVIEW

This section describes the software design of the CBARD Output Dissemination Software (SCI-OUT). The function of SCI-OUT is to disseminate CBARD outputs to non-FAA users via the CBARD User Network. SCI-OUT interfaces with SCI-SYS software on the CBARD Radar Network.

8.2 ARCHITECTURAL DESIGN

8.2.1 SCI-OUT Components

SCI-OUT consists of two software components, CBOutputRecv and NBCMsgServer.

CBOutputRecv

CBOutputRecv performs the following functions within SCI-OUT:

- Receives aerosol release detection, aircraft location, and system status messages from SCI-SYS and forwards these messages to the NBCMsgServer component of SCI-OUT and also to SCI-RMON.
- Sends SCI-OUT status messages to SCI-SYS, so that health of the output dissemination equipment can be monitored by the CBARD SD operator.

NBCMsgServer

NBCMsgServer performs the following functions within SCI-OUT:

- Receives aerosol release detection and system status messages from CBOutputRecv.
- Generates a NBC-1 format XML text message for each aerosol release detection message.
- Generates NBC-1 format XML text message each time a system status message indicates that the CBARD mode for a particular site has been activated or deactivated.
- Maintains a list of destination addresses that correspond to an EOC facility for each CBARD site.
- Sends NBC-1 format aerosol release warning and CBARD mode activation/deactivation messages to the destination addresses corresponding to the specified CBARD site.

- Sends a status message to CBOutputRecv after each attempted transmission to a destination EOC system, indicating whether or not the destination address was reachable.

8.2.2 Interface Design

Interface Identification and Diagrams

Figure 8-1 illustrates the software interfaces for SCI-OUT.

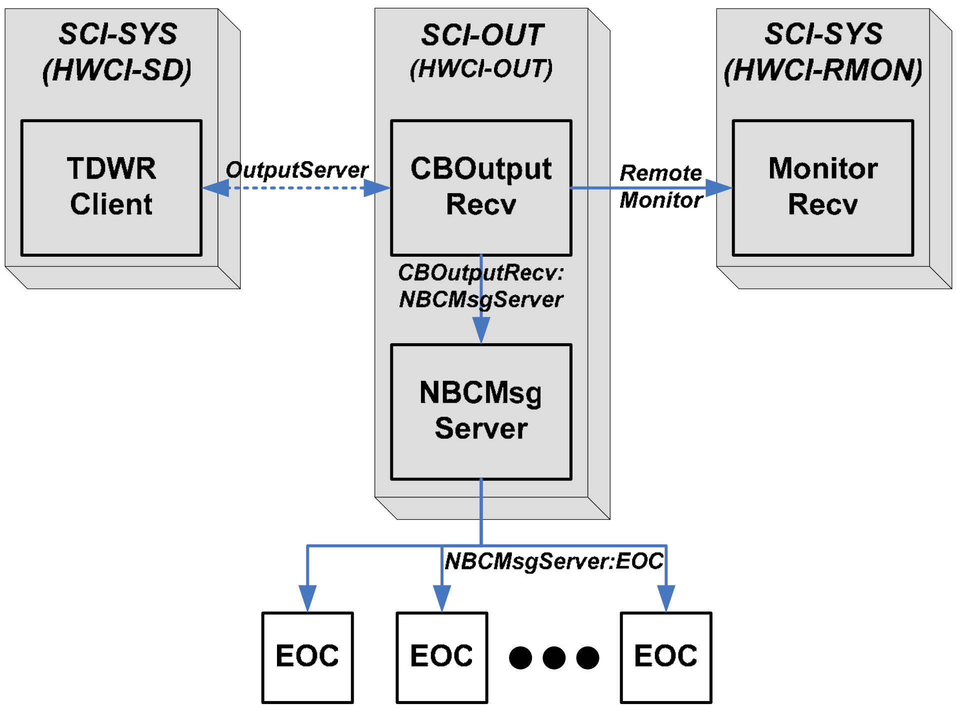

Figure 8-1. SCI-OUT software interfaces.

OutputServer Interface

This paragraph describes the interface between the SCI-SYS TDWRClient software and CBOutputRecv. CBOutputRecv receives System Status (CBStatus) and Aerosol Release Detection (CBDetection) messages from SCI-SYS. CBOutputRecv sends System Status (CBStatus) messages from SCI-SYS. There are three types of CBStatus messages sent to SCI-SYS, called CBOutputRecv Status messages, NBCMsgServer Status messages, and EOC Status messages. Table 8-1 provides references for the message format details.

TABLE 8-1

CBStatus Message Types Sent To SCI-SYS

Record Name	Comments	Format Reference
CBAircraftLoc	Location of aircraft as reported by the ASR-9 radar.	Table A-1
CBDisable	Disable CBARD mode.	Table A-3
CBEnable	Enable CBARD mode.	Table A-4

The following status messages are processed by SCI-OUT:

- Status messages originated by the TDWR site that indicate the state of the TDWR and associated SCI-TDWR.
- Status messages originated by the CBOutputRecv (SCI-OUT) software once every 30 seconds that indicate that the OutputServer interface (described below) is functioning properly.
- Status messages originated by the NBCMsgServer (SCI-OUT) software once every 30 seconds that indicate that NBCMsgServer is functioning properly.
- Status messages originated by the NBCMsgServer (SCI-OUT) software after each attempt to transmit a plume detection or status message to the NBCAnalysis software located at an end-user EOC facility.

CBOutputRecv:NBCMsgServer Interface

This paragraph describes the interface between CBOutputRecv and NBCMsgServer. CBOutputRecv sends System Status (CBStatus) and Aerosol Release Detection (CBDetection) messages to NBCMsgServer. NBCMsgServer sends System Status (CBStatus) messages to CBOutputRecv. There are two types of CBStatus messages sent from NBCMsgServer to CBOutputRecv, called NBCMsgServer Status messages and EOC Status messages.

NBCMsgServer:EOC Interface

This paragraph describes the interface between NBCMsgServer and the HDCBU EOC NBCAnalysis software that is external to CBARD. NBCMsgServer outputs Plume Detection and CBARD mode change messages to the NBCAnalysis software running on the HDCBU EOC computer. The output is sent to a predefined URL-type address containing a specific TCP/IP port on the external computer that is connected to the CBARD User Network.

9. PROGRAM STATUS AND FUTURE WORK

9.1 PROGRAM STATUS

A working CBARD prototype test bed was developed during fiscal year 2004. This system was deployed in Oklahoma City at the Mike Monroney Aeronautical Center (MMAC) and Oklahoma City Rescue 911 Center. The test bed utilized a non-operational TDWR located at the Program Support Facility (PSF) on the MMAC campus. Operational Oklahoma City ASR-9 aircraft surveillance data was obtained by a real-time feed from an FAA Academy facility on the MMAC campus. The test bed utilized a private frame-relay WAN purchased by MIT Lincoln Laboratory.

The capabilities of this system were developed, tested, and demonstrated during several field tests conducted in Oklahoma City during fiscal year 2004. This effort culminated in an end-to-end technical test in May, and an Operational Demonstration (OD) in August, 2004. An independent group of experts observed these tests and published measurements of system performance in terms of probably of detection and false alarm rate.

As a result of the OD testing, a few software problems were discovered and were subsequently corrected and tested using both the test bed in Oklahoma City and a test bed laboratory set up at MIT Lincoln Laboratory. The test bed laboratory was set up in early fiscal year 2005, in order to make final preparations for operational deployment of the CBARD prototype.

On 27 September 2004, the FAA approved a Test National Change Proposal (NCP) that would allow the CBARD prototype to be installed in Oklahoma City at the operational TRACON and TDWR facilities, with the Command Center to be located at the FAA Washington Operations Command Center (WOCC) at FAA Headquarters. The approval was for a 12 month period, and would require renewal after 27 September 2005. The NCP approval was conditional on obtaining explicit approval from FAA Vice President level managers. In January, 2005, the necessary management approval was imminent. Plans were also underway to seek a second Test NCP for an installation in the National Capitol Region (NCR).

The Army/DoD JPM NBC CA program office had no funding for the program for fiscal year 2005 and beyond. Attempts to obtain sponsorship within the Department of Homeland Security (DHS) were unsuccessful. As a result the Joint Program Executive Office for Chemical/Biological Defense (JPEO-CPD) has informed the FAA that no installation will occur unless such funding can be obtained.

9.2 FUTURE WORK

9.2.1 Roadmap for CBARD Enhancements

At the beginning of FY 2005, a roadmap was developed for future enhancements to the CBARD system. The initial CBARD capability and its associated software were named version 1. Two additional functional enhancement milestones were envisioned, called version 2 and 3, respectively.

CBARD version 2 would include the following:

- Address known algorithm and software deficiencies:
  - Eliminate processing of invalid TDWR data upon activation of CBARD mode.
  - Eliminate the use of the "Rolling Ball" reflectivity feature detector.
  - Fix "M-of-N" alert reporting filter bugs.
  - Eliminate the use of the 360 degree scan strategy algorithm.
  - Correct occasional erroneous track velocity data in the aircraft location cue message.
- Implement Operator Display feature enhancements recommended during the FAA Human Factors Evaluation process.
- Support simultaneous activation of CBARD mode at two TDWR facilities.
- Improve aircraft tracking scan strategy and aircraft and plume detection algorithms.

CBARD version 3 would include two major enhancements (that will be discussed in more detail later in this section):

- CBARD Surveillance Mode.
- TDWR signal processing algorithm enhancements.

The rest of this section discusses the enhancements listed above in greater detail.

9.2.2 Operator Display Modifications

In FY 2004, the FAA established a CBARD User Working Group, consisting mainly of current or former air traffic controllers and managers. A Human Factors specialist gathered their inputs, and wrote a document recommending changes and enhancements to the display software. A brief discussion of these proposed changes follows. In addition, FAA had planned to conduct a Human Factors Evaluation with some of the actual people would be CBARD operators at the WOCC.

1. Track Information: Remove the designated track number (DTN) field from the track data block. Move the DTN to the hooked track tag information in the upper left corner when a track is selected. Remove the rollover data tag information presented in white text next to the aircraft when the operator slews on an aircraft. Change the range and bearing information to show the bearing and DME from the cursor/arrow location to the selected aircraft.

2. Map Overlays: Add overlays for prominent areas, such as Disney World. Add county boundary overlays at lower zoom levels. Change the label for political boundaries from "PolB" to "States."

3. Filtering: Change the label "no beacon" to "primary." Allow multiple filters to be selected at the same time.

4. Redundant Coding for CBARD Detected Release: Add an aural alarm if an aerosol release is detected. In addition, make the red circle around the aircraft track icon blink for 10 seconds.

5. Display Detection Area: After the detection of a release, the spot of the release should be marked, and a manual command would be used to remove the designation.

6. History Trails: Add a feature that allows the length of the history trail to be changed, including support for full history trail, 5 minutes, 2 minutes, or none.

7. Depiction of CBARD Coverage Area: Add an overlay to show the effective CBARD detection coverage area.

8. Range Rings: Allow the WOCC operator to change the reference point for range rings.

9. Selection Buttons: Make the buttons for filtering and overlay toggles provide clear feedback to indicate whether or not the buttons are toggled on.

10. Site Home Buttons: Remove the site home buttons in the lower right menu. Modify the site home buttons on the CBARD Status Display Panel so that they indicate when the mouse cursor is scrolled over them.

11. Display Color Palette: The current display colors are not acceptable. The brightness control needs to be adjusted. The default setting should be mid-range of brightness and then be adjusted to either brighter or dimmer. The recalibrated brightness control should allow nothing darker than the presently obtainable brightest value. The colors of the buttons and map overlays should be revised.

12. Keyboard: Remove the need for a keyboard. Functionality such as range rollover that currently requires use of the shift key would have to be revised so that a mouse or track ball would be sufficient to execute the commands.

9.2.3 Simultaneous Activation of CBARD mode at Two TDWR Facilities

CBARD version 1 has the limitation that it can only activate one TDWR into CBARD Tracking Mode at a time. A possible terrorist threat scenario would include simultaneous, coordinated attacks in multiple cities. CBARD version 2 would support simultaneous activation of CBARD Tracking Mode at two TDWR facilities.

Implementation of the simultaneous multiple CBARD enable feature requires software modifications to the Situational Display (SD) software and the System Control and Monitoring software.

A key issue to be considered is how to present this capability to the CBARD operator. When CBARD Tracking Mode is activated, the operator must monitor the system continuously. Interruptions in the suspect aircraft track can cause CBARD mode to be deactivated, and therefore require the operator to re-activate the system. The ideal monitoring set up would be separate display hardware for each site, or at least for each geographic region. Unfortunately, this is not practical for the FAA WOCC location, due to the cost of manpower to staff such an operation and physical space limitations for placing computer monitors. Another approach would be to run multiple copies of the SD software on the computer and have a single monitor. This may not be ideal from a human factors point of view. Another possibility would be to modify the display software to add a second track viewing window. This would at least handle the case of simultaneous monitoring of two cities. If CBARD were to be implemented nationwide, a strategy would have to be developed that would scale appropriately.

The CBARD User Working Group recommended that two copies of the CBARD should be displayed on one monitor. The two displays on one monitor could either share a split screen or the operator may toggle between the two. Testing would need to be conducted to determine the best method for this situation. If more than two TDWRs are used, then further investigation would be required for how to display CBARD.

9.2.4 Plume Detection Algorithm Improvements

Following the OD, a comprehensive failure analysis was performed by Lincoln Laboratory for cases identified by the Army. Through playback and review of the archived results, several predominant causes for false alarms, missed warnings, and poor aircraft tracking were identified. A summary of this analysis is provided in Appendix B. Several of these issues were flagged for remediation in CBARD version 2. Following are descriptions of the version 2 software modifications that have already been implemented (but not fully field-tested) as of this writing:

1. Processing of invalid TDWR data upon activation of CBARD mode.

Following activation of CBARD mode, there is a delay between completing the standard TDWR monitor mode scan pattern in progress and execution of the customized CBARD search scan pattern that is constructed for scanning the aircraft location based on the ASR-9 cue. During these mode switch-over delays, full or partial TDWR monitor-mode scans were often delivered to the detection algorithm (which had already received the CBARD enable message) and the algorithm would dutifully begin to process the TDWR data for aircraft and plume features. These remnant TDWR monitor mode scans were typically not at the intended altitude, and often contained low-altitude clutter residue features that were mistaken for aircraft targets when processed.

This problem has been fixed by modifying two processes. First, the TDWRServer program, which translates the TDWR base data and sends it to the detection algorithm, was modified to pass along the scan type information along with the reformatted radar data. Second, the detection software (ARDA) was modified to check the scan type and skip the aircraft and plume feature detection processing if the scan type is not "PPI_SCAN", which is the type associated with a CBARD scan.

2. Eliminate the use of the "Rolling Ball" reflectivity feature detector.

The "Rolling Ball" filter was one of two image pre-filtering techniques used to background-normalize a TDWR reflectivity image prior to aircraft feature detection. The rolling-ball filtered reflectivity image was used as input to one of the aircraft feature detectors (AircraftRollBallDzDetector). Prior to the OD, it was discovered that this filter did not respond properly to missing TDWR reflectivity values, especially when these missing data areas were surrounded by high-background reflectivity values. Bright peaks or holes were created in the rolling ball-filtered reflectivity image and these were subsequently misinterpreted by the aircraft feature detector, causing false aircraft detections that disrupted TDWR tracking of the aircraft.

The affected aircraft feature detector has been modified to use an alternative existing spatial window-averaged normalized reflectivity image that does not suffer from the above-mentioned artifacts. This window-averaged normalized reflectivity image was already in use by all of the other aircraft and plume feature detectors in the system. The modified aircraft feature detector has been renamed "AircraftReflPeakDetector".

3. M-of-N bug fix.

To reduce the likelihood of false alarms from isolated detections of transient features, an M-of-N alert reporting filter requires that candidate intermediate plume detections occur on at least M=2 of the previous N=4 scans before an alert will be issued. Following an issued alert, the intermediate detection counter (M) should be reset to zero. This reset was not occurring, so false alarms were more likely to occur following a valid alert, as a single subsequent candidate detection could more easily satisfy the 2-of-4 criteria by association with the detections from the prior alert. At least one false alarm during the OD was directly attributable to this bug, which has now been fixed.

4. Elimination of 360-degree scan.

The CBARD scan strategy generation algorithm currently produces 360-degree scans when the expected aircraft position, based on ASR-9 position and velocity plus accommodation for a trailing plume, results in a scan sector width that exceeds 180 degrees. In field tests prior to the OD, the 360-degree scan pattern was found to be too time consuming and nearly always resulted in loss of tracking. Changes were made to the CBARD scan strategy generation algorithm prior to the OD to eliminate the 360-degree scan pattern and use the standard 2-tilt elevation scan sequence instead at short range. At the request of the Army, these changes were uninstalled for the OD (to maintain consistency of behavior with prior tests), but should be re-installed for CBARD version 2.

Other failures and difficulties noted in the OD post-analysis are largely attributable to the inherently challenging nature of the detection problem, and not necessarily associated with coding errors or fundamentally incorrect algorithmic approaches. A number of potential improvements were in the early stages of investigation at the time of this report:

5. Adjustments of thin line suppression and frame-integrated plume detector parameters.

These adjustments include the decay rate of the suppressive interest generated by the thin line suppression detector, temporal averaging weights used in the frame-integrated plume detector, and the number of images retained in the image histories utilized by these detectors.

6. Use of dynamic “initial” plume detector scoring functions.

The initial plume detector identifies the streak of enhanced reflectivity in the initial release region immediately behind the aircraft. This detector could possibly be further optimized if it utilized a different set of pattern matching scoring functions for high and low background reflectivity conditions.

7. Use pixel-wise maximum of current and prior reflectivity images as input to frame-integrated plume detector.

The frame-integrated plume detector performs temporally weighted averaging of the signals over several scans in a rectangular frame region behind the aircraft in the radar reflectivity image. The oscillating 2-tilt elevation scan pattern that is utilized during CB mode often results in hitting the plume more solidly on every other scan. Naively including the “off-plume” scan in the average weakens the integrated result, making detection more difficult. Taking the pixel-wise maximum of the current and prior reflectivity images prior to averaging may lessen this weakening of the integrated signal.

In general, some degree of improved plume detection capability should be achievable by making modest algorithm software modifications and parameter adjustments. Some reportable plume signatures noted by observers were not detected by the current algorithm because the current strategy is to automatically reduce sensitivity in the presence of adverse background clutter conditions in order to reduce the chance of a false warning. The algorithm automatically reduces its detection sensitivity as a function of increasing mean background reflectivity and variance (i.e., “yellow” or “red” confidence categories). By improving the feature discrimination capability in the early processing stages, the ability of the algorithm to identify plume releases should be improved in the more challenging background clutter conditions.

The general future improvement strategy consists of:

- Improving the discrimination capability of the existing individual feature detectors. This is done by optimizing feature detector scoring functions and pattern matching template sizes as the database is increased.

- Enhancing the ability of the "look-ahead" thin line suppression feature detector to identify and flag pre-existing clutter that can cause false alarms

- Reducing the amount of adaptive desensitization to gain detections (after implementing the above two improvements), and making adjustments to incur an acceptable false alarm rate.

9.2.5 Aircraft Tracking Scan Strategy Algorithm Enhancements

A significant 3-DAT algorithm improvement would be to modify the scan strategy when the aircraft track is about to be lost. The idea is to avoid the need for the SEARCH mode scan strategy (see section 5.1.1). A scan strategy modification could be made as long as the aircraft is within some limited range of the last “good” detection. The software modifications would required three additional functions within 3-DAT. Firstly, we would use the previous detections from ARDA to determine that the track is poor (about to be lost) and switch to the modified scan strategy. Secondly, we would select the best elevation angles. Thirdly, we would determine when the aircraft detections are once again acceptable and switch back into the TRACK mode.

An initial version of the modified SEARCH mode was coded into the 3-DAT for offline testing. The algorithm was modified to switch from the TRACK mode to the SEARCH_MOD mode when three consecutive tilts from the TDWR produced a radar cross section (RCS) value for the aircraft below 0.025 m^2, which is roughly equivalent to a 30 dBZ return at 10 kilometers. The algorithm then generates a modified scan strategy that consists of two elevation scans, one located at a best guess of the aircraft’s altitude, and the other located at the last known aircraft altitude. The best guess elevation angle was based on the last two elevation scans and their respective elevation angles and radar cross section values. Evaluation of data from the CR7 field test indicated that the cropduster style aircraft has a radar cross section of 1.0 m^2. For every half beam width the radar is off from the true aircraft elevation the RCS value will drop by a factor of 10. A RCS value of 0.01 m^2 from a 1.5 degree elevation angle suggests that the aircraft is actually at either 1.0 or 2.0 degrees. Using the last two elevation angles and RCS value, the algorithm can find the right direction (above or below) to move from the last elevation angle. Finally, the algorithm would decide when to return to the TRACK mode or abandon the SEARCH_MOD mode and return to the basic SEARCH mode. The algorithm was modified to switch to TRACK mode when the last two elevation scans from the TDWR produced RCS values greater than 0.1 m^2. Once the two elevation scans produce high enough RCS values that the ARDA detection and the 3-DAT best guess are within 0.25 degrees, it is likely that the aircraft has stopped a descent/ascent, or the aircraft was been reacquired after several poor detections.

9.2.6 CBARD Surveillance Mode

CBARD version 1 allows TDWR to track a single aircraft. We will refer to this as CBARD Tracking Mode. The major limitation of Tracking Mode is that it requires timely and specific intelligence about an imminent threat.

CBARD Surveillance Mode would allow TDWR to track multiple aircraft within an operator defined region of interest (ROI). Thus, specific intelligence regarding an imminent threat would not be required. Rather than locating and selecting a specific aircraft track on the Operator Display, the operator would simply activate Surveillance Mode. Automated location cues would be provided for all aircraft within the ROI. For Surveillance Mode, a sector scan strategy would be devised that surveils elevation angles sufficient for the likely crop duster agent release scenarios. The Aircraft and Plume Detection algorithm would have to be modified so it could track multiple aircraft targets and look for aerosol plume signatures behind all of them.

If evidence of a release occurs during Surveillance Mode operation, it may be beneficial to automatically switch to Tracking Mode and look at that aircraft more closely.

9.2.7 TDWR Signal Processing Enhancements

Under FAA sponsorship, MIT Lincoln Laboratory is currently working on a signal processing hardware upgrade and algorithm software enhancement for the TDWR to improve performance for its weather detection mission. The upgraded TDWR signal processing system, with its new digital receiver hardware, will provide improved performance. The modern scalable platform will eventually include improved signal processing algorithm software to address data quality issues that can limit the effectiveness of the HDCBU TDWR algorithms. Algorithms under development will address long-standing weather radar data quality issues, including elimination of range folding from distant weather, elimination of Doppler velocity aliasing due to high wind speeds, and recovery of weather echoes in the presence of strong ground clutter returns. It also provides a hardware platform for implementing the Doppler Signature Enhancement algorithms proposed as follow-on work for the HDCBU program. An engineering prototype will be fielded at a single TDWR site during FY 2006. Full-scale production and deployment of the new Radar Data Acquisition (RDA) subsystem is scheduled to begin in FY 2007.

The upgraded TDWR signal processing system will provide a platform that will enable signal processing algorithm enhancements to be made for the HDCBU program. Such improvements would be aimed at making the aerosol release plume signatures more recognizable in the weather radar data. Algorithm improvements under consideration will improve the accuracy of the estimated parameters that are used by the current HDCBU CBARD algorithm. Doppler velocity is a measure of radar velocity, which means velocity with respect to the radar. The Doppler velocity signature of crop duster releases is often different from those of the surrounding air. Proposed algorithm enhancements would provide more accurate measurements that would be necessary to detect such differences.

The following algorithm enhancements could be made, and will improve the plume detection capability:

- Increase PRF from 1672 to 1930 Hz.
- Decrease azimuth dwell size from 1 deg to 1/3 deg.
- Filtered spectral width computation.

Currently the TDWR pulse repetition frequency has a hard-coded maximum of 1672 Hz for low elevation scans. This limit is imposed by the FAA range coverage requirement of 48 nautical miles. With full control of the software specification in the new system, it will be a trivial matter to increase the PRF to the transmitter limit of 1930 Hz, which will improve the accuracy of all estimated parameters (reflectivity, velocity, and spectral width).

The dwell size is also currently hard-coded. By decreasing it, the signal-to-noise ratio of plumes on radial runs will be increased. The improvement results because on radial runs the lateral angle subtended by the plume is much less than 1 deg. Therefore, the mean power of the plume computed over a dwell increases as the dwell angle is decreased, because the antenna beam spends less time looking away from the plume. The mean background signal is unchanged because it is volume filling. This enhancement in plume contrast with azimuthal resolution disappears as the plume angle tends toward a cross-beam orientation.

Field test results showed that the Doppler spectral widths of crop duster releases are sometime wider than those of the surrounding air. We suspect that this is due to wake turbulence behind the aircraft. However, the standard pulse-pair spectral width product output of the current TDWR is usually too noisy for the distinction to be seen. A filtered spectral width computation technique has been shown to enhance the contrast between plume and background spectral widths. This method can be implemented in the new signal processor. Items 2 and 3 are really key, because they aid in plume discrimination. Item 1 improves the overall data quality.

APPENDIX A
CBARD MESSAGE FORMATS

This appendix describes the detailed message formats for the TDWR CBARD system.

TABLE A-1

Aircraft Location Message Format

Data Element Name	Description	Data Type	Size (Bytes)	Units	Range of Values
recID	Record identification number	unsigned short	2		5815
recLen	Record length in 2-byte words	short	2		
seqNum	Unique record sequence number	unsigned int	4		
time	Record time stamp	short array	12	GMT (MDYhms)	
radarName	Radar Site ID	char	16		e.g., "OKC"
radarType	Radar Type providing surveillance data	char	16		e.g., "ASR-9"
radarLatitude	Radar Latitude	int	4	degrees north x 100000	0 – 36000000
radarLongitude	Radar Longitude	int	4	degrees west x 100000	0 – 36000000
radarAltitiude	Radar Altitude	short	2	Above MSL	
bcnTxCode	Beacon Transponder Mode 3/A code	unsigned short	2		
bcnAltitude	Beacon Altimeter Altitude	short	2	meters above MSL	
altitude	TDWR determined altitude of aircraft	short	2	meters above MSL	
xPos	Location east of radar	int	4	Meters east x 100	
yPos	Location north of radar	int	4	Meters north x 100	
speed	Track Speed	short	2	Meters/sec x 100	
heading	Track Heading	short	2	degrees	0 - 359
amplitude	TDWR-determined aircraft amplitude	short	2	dB x 100	
nCoast	TDWR coasted detection counter	short	2		
trackNum	Track ID Number from surveillance radar	unsigned short	2		1 - 65535
radarElev	TDWR elevation angle (if TDWR surveillance)	short	2	degrees x 100	0 - 36000

TABLE A-2

CBDetection Message Format

Data Element	Description	Data Type	Size (Bytes)	Units	Range of Values
recID	Record identification number	unsigned short	2		5820
recLen	Record length in 2-byte words	short	2		
seqNum	Unique record sequence number	unsigned int	4		
validTime	Record time stamp	short array	12	GMT (MDYhms)	
radarName	Radar Site ID	char array	16		e.g., “OKC”
radarType	Radar Type providing plume detection data	char array	16		e.g., “TDWR”
radarLatitude	Radar Latitude	int	4	degrees north x 100000	0 – 36000000
radarLongitude	Radar Longitude	int	4	degrees west x 100000	0 – 36000000
radarAltitiude	Radar Altitude	int	4	MSL	
ID	Detection ID number	int	4		
deltaT	Time difference from current time	int	4	seconds	= 0 current detection < 0 historical detection > 0 forecast
startX	Start X-coordinate of plume	int	4	meters from radar x 100	
startY	Start Y-coordinate of plume	int	4	meters from radar x 100	
endX	End X-coordinate of plume	int	4	meters from radar x 100	
endY	End Y-coordinate of plume	int	4	meters from radar x 100	
startLat	Start latitude of plume	int	4	degrees north x 100000	
startLong	Start longitude of plume	int	4	degrees west x 100000	
endLat	End latitude of plume	int	4	degrees north x 100000	
endLong	End longitude of plume	int	4	degrees west x 100000	
length	Length of plume	int	4	meters x 100	
sfcWindSpeed	Width of plume	int	4	meters x 100	
sfcWindDir	Area of plume	int	4	km^2 x 100	
orientation	Plume orientation	short	2	degrees from compass north	0 - 359
altitude	Altitude of plume	short	2	meters MSL	
duration	Event duration	int	4	seconds	
bckgrdMean	Background mean reflectivity	short	2	dBZ x 100	
bckgrdStdSDev	Background standard deviation	short	2	dBZ x 100	
windSpd	Ambient wind speed	short	2	knots	
windDir	Ambient wind direction	short	2	degrees	0 - 359

TABLE A-3

CBDisable Message Format

Data Element	Description	Data Type	Size (Bytes)	Units	Range of Values
recID	Record identification number	unsigned short	2		5821
recLen	Record length in 2-byte words	short	2		
seqNum	Unique record sequence number	unsigned int	4		
time	Record time stamp	short array	12	GMT (MDYhms)	
radarName	Radar site ID	char array	16		e.g., "OKC"
spare	Reserved for future use	char array	4		

TABLE A-4

CBEnable Message Format

Data Element	Description	Data Type	Size (Bytes)	Units	Range of Values
recID	Record identification number	unsigned short	2		5822
recLen	Record length in 2-byte words	short	2		
seqNum	Unique record sequence number	unsigned int	4		
time	Record time stamp	short array	12	GMT (MDYhms)	
radarName	Radar site ID	char array	16		e.g., "OKC"
spare	Reserved for future use	char array	4		

TABLE A-5

CBScanStrategy Message Format

Data Element	Description	Data Type	Size (Bytes)	Units	Range of Values
recID	Record identification number	unsigned short	2		5825
recLen	Record length in 2-byte words	short	2		
seqNum	Unique record sequence number	unsigned int	4		
issueTime	Record time stamp	short array	12	GMT (MDYhms)	
numValidScans	Number of elevation scans in strategy	unsigned short	2		1 - 30
startMode	When TDWR should start using this scan strategy?	unsigned short	2		0 = start immediately 1 = start after end of current elevation scan 2 = start after end of current volume scan
The remainder of the message is an array of 30 scan definitions, each with the format shown below. The field 'numValidScans' (above) specifies how many of the 30 scan definitions contain valid elevation scan data.					
startAzimuth	Starting azimuth	short	2	degrees	0 - 359
endAzimuth	Ending azimuth	short	2	degrees	0 - 359
elevation	Scan elevation	float	4	degrees	-1.0 – 60.0
rotationRate	Antenna scan rate	float	4	deg/sec	-30.0 – 30.0
prf	Pulse Repetition Interval Value maps to standard TDWR PRI table: 518, 538, 558, 578, 598, 618, 638, 658, 678, 698, 718, 738, 758, 778, 798, 818, 838, 858, 878, 898, 918, 938, 3066	int	4		1 – 23

TABLE A-6

FrameStart Message Format

Data Element	Description	Data Type	Size (Bytes)	Units	Range of Values
recID	Record identification number	unsigned short	2		1470
recLen	Record length in 2-byte words	short	2		
seqNum	Unique record sequence number	unsigned int	4		
frameType	Arbitrary identifier	int	4		
frameNum	Sequence number	int	4		
timestamp	Record time stamp	short array	12	GMT (MDYhms)	

TABLE A-7

FrameEnd Message Format

Data Element	Description	Data Type	Size (Bytes)	Units	Range of Values
recID	Record identification number	unsigned short	2		1471
recLen	Record length in 2-byte words	short	2		
seqNum	Unique record sequence number	unsigned int	4		
frameType	Arbitrary identifier	int	4		
frameNum	Sequence number	int	4		
timestamp	Record time stamp	short array	12	GMT (MDYhms)	

TABLE A-8

SWStatus Message Format

Data Element	Description	Data Type	Size (Bytes)	Units	Range of Values
recID	Record identification number	unsigned short	2		14111
recLen	Record length in 2-byte words	short	2		
seqNum	Unique record sequence number	unsigned int	4		
deviceId	Hardware device ID	short	2		
moduleId	Software module ID	short	2		
genericStatus	Status bits	short	2		0 = OK Else TBD bit pattern
cpuUsage	Reserved for future use	short	2		

TABLE A-9

CBStatus Message Format

Data Element	Description	Data Type	Size (Bytes)	Units	Range of Values
recID	Record identification number	unsigned short	2		5823
recLen	Record length in 2-byte words	short	2		
seqNum	Unique record sequence number	unsigned int	4		
validTime	Record time stamp	short array	12	GMT (MDY hms)	
radarName	Radar site ID	char array	16		e.g., “OKC”
moduleId	Software module ID of message originator	short	2		1=LLOT 2=3DAT 3=Base Data Server 4=ARDA Server 5=ARDA 6=RPG 7=NBCMsgServer 8=CBOutputRecv
status	CBARD status for this site	unsigned short	2		0=Available (disabled) 1=Unavailable 2=Enabled
errCode	Error code	unsigned int	2		0x1=No error 0x2=TDWR Comm 0x4=TDWR Haz Mode 0x8=ARDA S/W Comm 0x10=3DAT S/W Comm 0x20=Status Fail 0x40=Status timeout 0x80=Status checksum error 0x100=NBCMsgServer Comm 0x200=NBCMsgServer send to EOC failure
confidence		unsigned short	4		
spare	Reserved for future use	char array	2		

APPENDIX B
HDCBU OPERATIONAL DEMONSTRATION FAILURE ANALYSIS

This appendix summarizes the predominant causes of failure identified by analyses that were conducted by Lincoln Laboratory following the HDCBU Operational Demonstration (OD) in August, 2004 at Oklahoma City. For each identified cause, a list of OD case numbers for which the particular cause was noted is given, followed by a description of the problem, recommendations for mitigating the failure, and status on implementing the recommendations. The summaries are organized following the OD failure categories defined by the Army:

1. Tracking
2. Missed warnings
3. False warnings
4. Communications
5. Software

B-1. PREDOMINATE CAUSES OF POOR AIRCRAFT TRACKING

B-1.1 ARDA processed remnant TDWR monitor mode data immediately after receiving CBEnable message.

Cases: 1,14,21,23,25,29,31,33

Description:

Remnant TDWR monitor scans were often received and processed by the detection algorithm immediately after CB mode activation. These scans often contained clutter peaks mistaken for aircraft.

This problem was exacerbated by the non-standard TDWR monitor mode pattern used during the first half of the OD. The special high vertical resolution, low altitude monitor scan pattern resulted in increased opportunity for receiving a clutter-contaminated scan.

Recommendations:

Modify tdwrServer and ARDA to recognize and avoid processing of non-CB mode scans:

1. Modify tdwrServer to pass along the TDWR scan type in the header or comment fields of the reformatted data that it sends to ARDA.

2. Implement additional logic in ARDA to skip aircraft/plume detection processing stages if a non-CB mode scan is received

Status:

Completed, with some limited field testing at OKC site.

B-1.2 ARDA detection of features too distant from ASR-9 cue location.

Cases: 1,14,21,23,32,33

Description:

The maxAircraftCueDistM radius parameter of 2 km is too large.

Recommendations:

Reduce maxAircraftCueDistM parameter from 2 km to 1 km.

Status:

This is a pre-existing algorithm parameter. No software modification needed. Parameter change has been tested against CR6 and CR7 datasets with no apparent loss of valid aircraft targets. Exercised in real-time during CR8 which had very good tracking results.

B-1.3 ARDA detection of features having improbable motion.

Cases: 14,23

Description:

Clutter tracking in these cases produced erratic aircraft motion vectors with small velocities. This information can be used to reject false aircraft targets.

Recommendations:

- Increase minAircraftSpeedMPS parameter from 0 m/s to 15 m/s (~30 kts).

- Implement additional logic to detect erratic motion vectors, and drop track and go back into search mode to re-acquire true aircraft.

Status:

- minAircraftSpeedMPS is a pre-existing algorithm parameter; no software modification needed. Parameter change has been tested against CR6 and CR7 data sets with no apparent loss of valid aircraft targets. Exercised in real-time during CR8.

- Additional logic to respond to erratic aircraft motion vectors has not been developed.

B-1.4 ARDA detection of features having low reflectivity or low radar cross section.

Cases: 14,29,32

Description:

Many of the false aircraft detections were on clutter residue having reflectivities or radar cross sections (RCS) that are smaller than true aircraft targets.

Recommendations:

Implement intelligent thresholding scheme to recognize repeated detections with RCS or reflectivity values below expected values. Drop false track and go back into search mode to re-acquire true aircraft.

Status:

Not implemented. Some investigative analyses have been conducted. See Figure 1 which plots RCS as a function of the difference in elevation angle between the actual (beacon) aircraft position and the radar-detected position for CR7 release trials. False detections due to clutter often occur at elevations substantially different from the true aircraft elevation. The figure suggests that for such large elevation differences often associated with false detections, the RCS is substantially less than when the true aircraft target is being more directly scanned during optimal tracking.

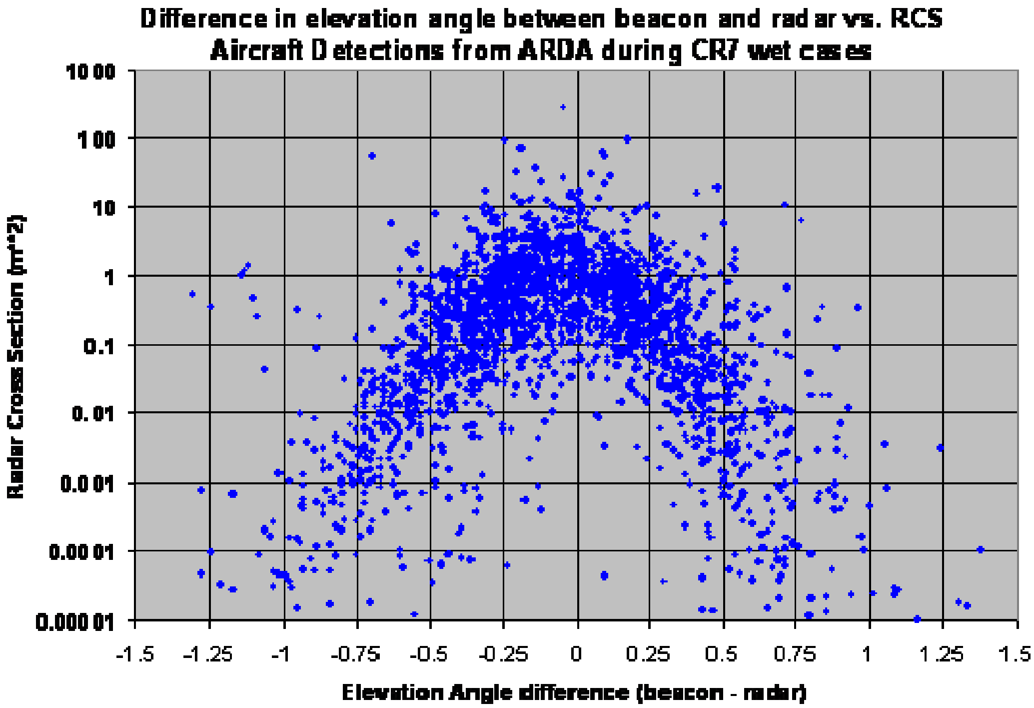

Figure B-1. Radar cross section (RCS) vs. deviation angle difference between true (beacon) location and radar detected location.

B-1.5 "Rolling ball" reflectivity filter bug

Cases: 32

Description:

This bug was reported prior to the OD. The rolling ball filter is one approach for normalizing the mean background reflectivity in the reflectivity image prior to detector processing. Missing input reflectivity image values (due to TDWR data quality editing, low SNR, etc.) are not properly handled by this filter, especially in high background conditions, resulting in holes or bright peaks in the image that are misinterpreted by the aircraft feature detectors.

Recommendations:

Only one aircraft feature detector currently uses the rolling ball filtered reflectivity. All other aircraft and plume detectors use an alternative normalization filter that is not adversely affected by missing data. This aircraft reflectivity detector should be modified to use the alternative normalized reflectivity as input.

Status:

This change has been implemented and tested against valid aircraft targets in datasets CR5 through CR8. Several false aircraft detections were eliminated. Additional testing with Cessna or targets of opportunity may be desirable.

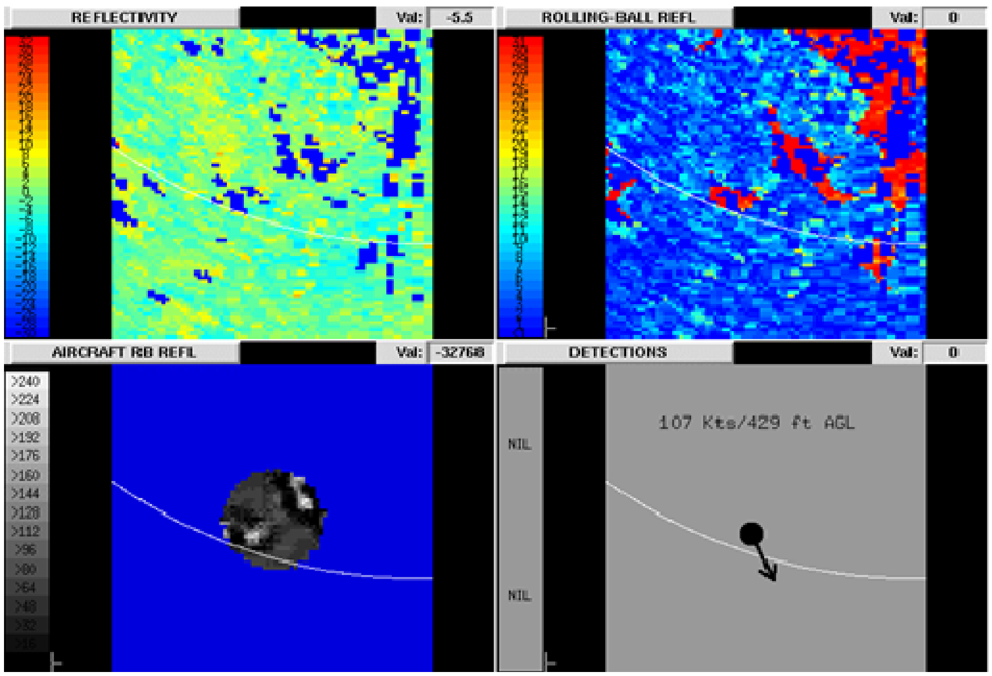

Figure B-2. Example of rolling ball filter bug causing false aircraft detection. The aircraft reflectivity detector in the lower left is processing the rolling ball filtered reflectivity in the upper right which has anomalous peaks (red) in areas of missing data. Data are from CR8.

B-1.6 360-degree CB mode scans at short range caused loss and prevented re-acquisition of aircraft.

Case: 31

Description:

The CBARD scan strategy generation algorithm currently produces 360-degree scans when the expected aircraft position (based on ASR-9 position and velocity) plus accommodation for a trailing plume, results in a sector width that exceeds 180 degrees. The 360-degree scan pattern has been found to be too time consuming and nearly always results in loss of tracking. In case 31, an anomalous aircraft speed issued by the CBARD tracker threw the TDWR into 360 mode.

Recommendations:

The 360-degree scan strategy should be eliminated and the standard 2-tilt elevation scan sequence should continue to be utilized at short range.

Status:

This change was implemented prior to and in effect during CR8. It was undone for the OD to ensure comparable behavior to CR7 for combining of results from the two datasets.

B-2. PREDOMINANT CAUSES OF MISSED WARNINGS

B-2.1 Background was too high and/or variable.

Cases: P8,9,11,P11,P12,31,P16,P17,39,44,53

Description:

Background was category YELLOW (11,31,39,44) or RED (P8,9,P11,P12,31,P16,P17, 53), with little or no observable plume signatures. Some of these cases (9,11,44) also suffered from too few scans of the plume due to tracking difficulties.

Recommendations:

None. Insufficient signal-to-background clutter to permit detection.

B-2.2 Tracking problems

Cases: P6,14,29,31,32

Description:

Aircraft was not being tracked, so release was not observed.

Recommendations:

None. See recommendations related to improving aircraft tracking.

B-2.3 Poor tracking without aircraft transponder

Cases: P5, P6

Description:

Unassisted (i.e. without beacon) CBARD search mode was configured to scan up to 3200' in 3-DAT scan strategy generator. Therefore, TDWR did not scan high enough and could not acquire aircraft without transponder when flying at 3500' or 4000'.

Recommendations:

Scan strategy search mode parameters should have been configured for expected altitude coverage.

Status:

For operational systems, desired altitude coverage needs to be clearly defined so search mode altitude parameters can be properly configured.

B-2.4 Poor plume scanning

Cases: 9,11,19,44,52

Description:

Aircraft maneuvering, e.g., rapid elevation changes, high altitude flight, made it difficult to choose scans that kept the aircraft in view while simultaneously providing good scan coverage of the trailing plume.

Recommendations:

When possible, flights below 2500’ should result in improved plume scanning.

Status:

No further actions proposed at this time.

B-2.5 Sub-optimal feature detector tunings

Cases: 34,52

Description:

The look-ahead thin line suppressor occasionally over-reacted to non-linear features, interfering with detection (case 34). Frame-integrated plume detector was slow to respond to brief plume evidence (case 52).

Recommendations:

Thin line suppressor parameters should be optimized for more accurate discrimination of linear reflectivity features. May need to adjust temporal averaging weights utilized by frame-integrated plume detector to make it more responsive.

Status:

Alternative plume detector tunings including the thin line suppressor are presently being investigated. No work has been done on adjusting the frame-integrated plume detector temporal averaging weights.

B-3. PREDOMINANT CAUSES OF FALSE WARNINGS

B-3.1 Sub-optimal response of feature detectors

Cases: 6,45,52,53

Description:

Two of the primary plume feature detectors -- the "Initial" and "Frame" detectors – appeared over-responsive to non-linear echoes inconsistent with true plume signatures.

Recommendations:

Parameters defining the scoring functions and templates used by the Initial and Frame plume detectors need to be re-examined in light of the expanded database provided by the OD cases.

Status:

Alternative scoring functions are presently being tested for the Initial thin line plume detector. Others are TBD.

B-3.2 Insufficient look-ahead identification of pre-existing clutter features by thin line suppressor.

Cases: 3,14,38,

Description

Two deficiencies have been identified:

1. The Initial and Frame plume detectors were supposed to delay their processing for 3 scans following a CB enable to allow time for the look-ahead thin line feature suppressor to identify pre-existing line features ahead of the aircraft. Due to an implementation error (bug), the delay counters for these two plume detectors were not being reset following each event, so this protective feature was only in effect following a system restart.

2. A fade factor parameter used to gradually decay the disconfirming interest regions identified by the thin line suppressor may be set too fast. The disconfirming interest regions need to persist long enough to allow time for the aircraft to fly through them so that they can counteract any false confirming evidence that may be generated behind the aircraft in the plume detection region.

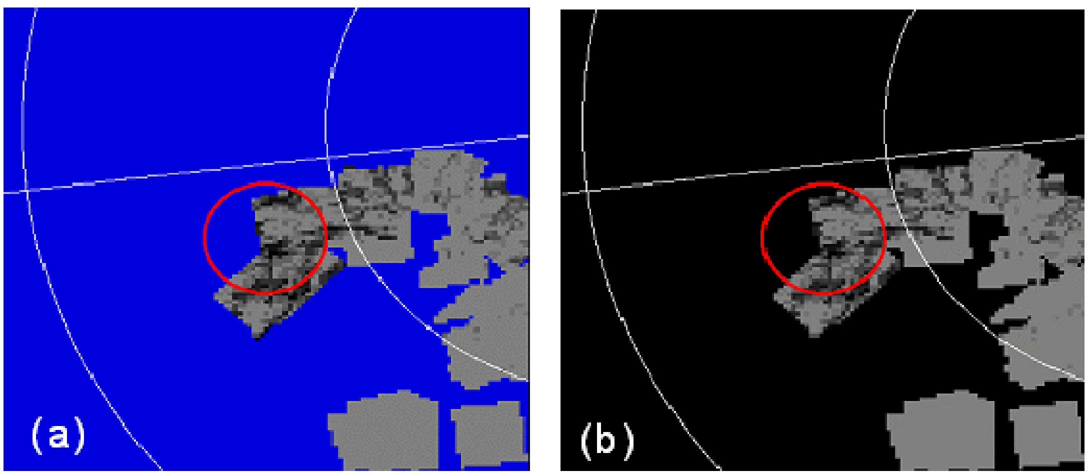

Figure B-3. Thin line suppression interest image for case E14 before and after initialization changes. (a) Original image with unprocessed regions set to nil (blue areas) (b) After software modification to set unprocessed regions to zero (black areas).

Recommendations:

1. The bug that prevented the intended 3-scan processing delay for the Initial and Frame plume detectors is easily fixed. However, this fixed processing delay approach has a fundamental weakness in that the amount of delay actually needed depends on the speed and trajectory of the aircraft. An alternative approach has been developed wherein the entire thin line suppression image is initialized with strong disconfirming interest everywhere (instead of the neutral values currently used). Then, as aircraft tracking proceeds, the look-ahead detector can undo or maintain the disconfirming interest as appropriate. Figure 3 shows the effects of this modification on the thin line suppression interest image.

2. Reduce the fade rate factor used to decay the disconfirming interest generated by the thin line suppressor.

Status:

Both of these recommended changes have been developed and have been validated by testing with existing data sets.

B-3.3 Precipitation echoes

Cases: 3,53

Description:

False warnings were generated on edges of precipitation and 2^{nd} trip (range-folded) weather echoes.

Recommendations:

ARDA needs to identify and mask precipitation and 2^{nd} trip echo regions from processing by the plume feature detectors. Improved plume detector discrimination may also be needed (See section 3.1).

Status:

TBD

B-3.4 M-of-N reset bug

Case: 3

Description:

The M-of-N alert reporting filter requires that intermediate plume detections occur on at least M=2 of the previous N=4 scans before an alert will be issued. Following a reported alert, the intermediate detection counter (M) should be reset to zero. This was not occurring during the OD, so a single, spatially correlated intermediate detection occurring within a 4-scan interval following a reported alert would immediately satisfy the M-of-N condition and be reported as a separate alert.

Recommendations:

M-of-N intermediate detection counters need to be reset following a reported alert.

Status:

Bug fix completed and tested with playback data. Not installed at this time.

B-4. COMMUNICATIONS FAILURES

B-4.1 NBCMsgServer radar name configuration problem prevented sending of alerts to EOC prior to E4 on 08/12.

Case: P3

Description

The NBCMsgServer reformats received plume alert messages into XML-formatted NBC messages that are sent to the EOCs. A configuration file contains the mappings of TDWR radar names to destination EOC URLs. Prior to case E4, an incorrect radar name ("PSF" vs "OKC") was listed in the site configuration file, so no match was found with the radar name contained in the incoming plume alert messages, and NBC messages were not sent.

Recommendations:

Radar name parameter in NBCMsgServer configuration file needs to be consistent with radar identifier contained in plume alert messages generated by ARDA.

Status:

This was fixed during the OD prior to case E4.

B-4.2 Data message corruption introduced by serial link between TDWR and CBARD Command Center.

Cases: P9,30

Description:

There were several instances in which messages sent from the TDWR to the CBARD Command Center software were found to have corrupted data.. This caused the downstream NBCMsgServer process to abort or hang.

Recommendations:

Replace serial link between TDWR and CBARD Command Center with Ethernet TCP/IP link.

Status:

Completed.

B-4.3 CBARD display software did not handle communication errors properly.

Case: P4

Description:

WOCC Operator Display EOC communications errors status did not work as intended during case P4. Lost connection to EOC was not indicated.

Recommendations:

Display and NBCMsgServer software must be modified so that EOC communication errors persist on the status display.

Status:

NBCMsgServer has been modified to output a status message every time an alert message is posted to an EOC, not just when a post fails. Corresponding changes have been made in the display software to differentiate between successful or failed post attempts as indicated in the status messages received from NBCMsgServer.

B-5. SOFTWARE FAILURES

B-5.1 System refresh button was red before first trial of the day.

Case: P8

Description:

WOCC Operator Display software problem has been duplicated using playback of recorded ASR-9 data.

Recommendations:

Display software must be modified to fix this problem.

Status:

TBD

B-5.2 TDWR automatic reconfiguration interfered with CBARD operation.

Case: 35

Description:

TDWR equipment "reconfiguration" protocol interfered with HDCBU system operation.

Recommendations:

FAA has already implemented a software modification to disallow HDCBU operation during TDWR maintenance protocols.

Status:

Completed.

APPENDIX C
CBARD SOURCE CODE ORGANIZATION

Figure C-1 provides a CBARD source code tree.

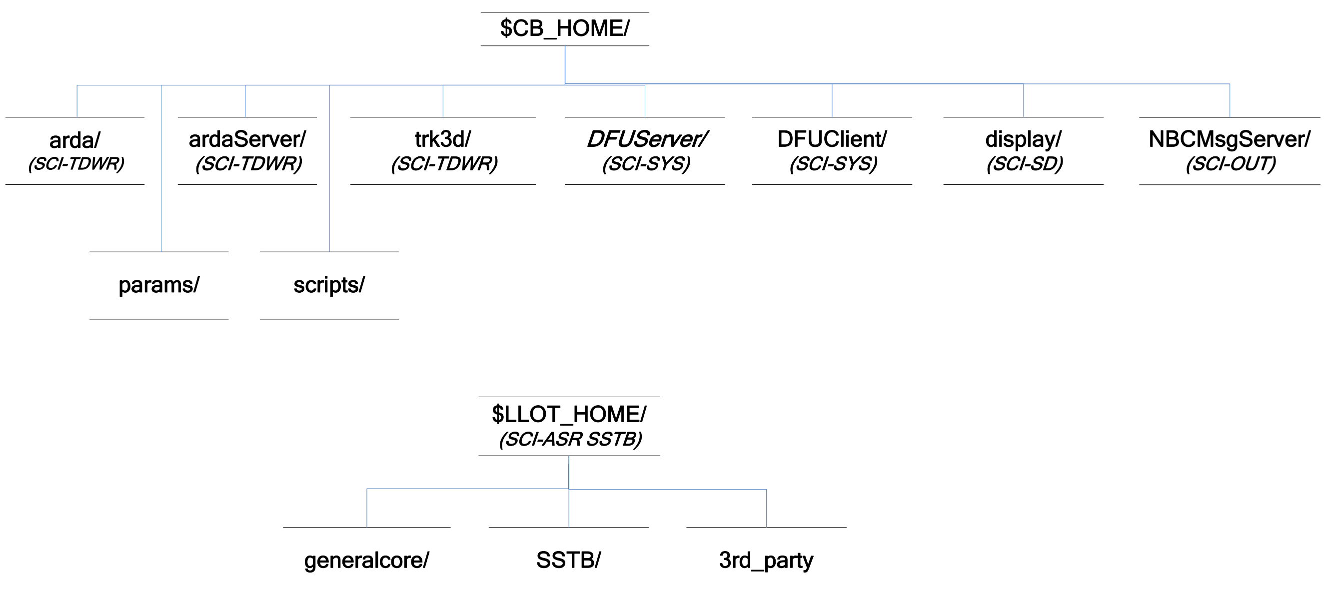

Figure C-1. CBARD source code directory structure.

GLOSSARY

9PAC	ASR-9 Processor Augmentation Card
ADW	Andrews Air Force Base
AF	Airways Facilities
AFB	Air Force Base
ARDA	Aerosol Release Detection Algorithm
ASIS	ASR-9 S Interface S
ASR-9	Airport Surveillance Radar Model 9
ATC	Air Traffic Control
ATO	Air Traffic Organization
CB	Chemical or Biological
CBARD	C-Band Aerosol Release Detector
CHI	Computer Human Interface
CI	Configuration Item
CORBA	Common Object Request Broker Architecture
COTS	Commercial Off-the-Shelf
DoD	Department of Defense
EOC	Emergency Operations Center
EPA	Environmental Protection Agency
FAA	Federal Aviation Administration
HDCBU	Homeland Defense Chemical Biological Umbrella
HWCI	Hardware Configuration Item
JPEO-CBD	Joint Program Manager for Chemical and Biological Defense
JPM NBC CA	Joint Program Manager for Nuclear, Biological, and Chemical Contamination Avoidance
LLFUSE	Lincoln Laboratory Fusion Tracker
LLOT	Lincoln Laboratory Optimized Tracker

MIT/LL	Massachusetts Institute of Technical Lincoln Laboratory
OKC	Oklahoma City
PCI	Portable Computer Interface
PSF	Program Support Facility
RMMS	Remote Maintenance Monitoring Subsystems
RPG	Radar Product Generator
SCI	Software Configuration Item
SCIP	ASR-9 Sensor C Interface Processor
SD	Situation Display
SDN	Surveillance Data Network Software
SIU	Sensor Interface Unit
SLEP	System Life Extension Program
SSDD	System/Subsystem Design Description
SSTB	Surveillance System Test Bed Software
TACSIT	Tactical Situation Display
TDWR	Terminal Doppler Weather Radar
TRACON	Terminal Radar Approach Control Facility
UPS	Uninterruptible Power Supply
WOCC	Washington Operations Command Center

REFERENCES

[1] NAS-IC-34032105, U.S. Department of Transportation Federal Aviation Administration Interface Control Document for Airport Surveillance Radar Model 9 (ASR-9) / Standard Terminal Automation Replacement Systems (STARS), 14 June 2001.

[2] "Interface Design Document (IDD) for the Network UNIO (NUNIO) Module to Host System Interface", Sensis Corporation, 8 September 2003. Document number: 790-003460, version 21, CAGE code 1EG52.

[3] "ASR-9 Processor Augmentation Card (9-PAC) Phase II Scan-Scan Correlator Algorithms", MIT Lincoln Laboratory Project Report ATC-298, R.D. Grappel, 26 April 2001. Lexington, MA.

[4] "FAA-E-2716, Specification Mode Select Beacon System."

[5] "Taskrunner: A Method For Developing Real-Time System Software", L.M. Hebert, 2002 High Performance Embedded Computing Workshop, MIT Lincoln Laboratory.

[6] Personal communications with Dr. J.L. Gertz, Technical Staff, Air Traffic Control Systems Group at MIT Lincoln Laboratory.

[7] Personal communications and email correspondence with T. Roe, Technical Staff, Air Traffic Control Systems Group at MIT Lincoln Laboratory.

[8] "Integrated Terminal Weather System (ITWS) TDWR/ITWS-PG-ICD (Part 1)", NAS-IC-31025214, Cage Code 49956, 17 May 2000, prepared by Raytheon Systems Company, Sudbury, MA.

[9] "WSP Utility Libraries", MIT Lincoln Laboratory Project Report ATC-284, O. Newell, 23 October 2000. Lexington, MA.

[10] "A Coordinate Conversion Algorithm for Multisensor Data Processing," MIT Lincoln Laboratory Project Report ATC-139, E.M. Shank, 5 August 1986.