

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, CA 94035-1000

UAS-NAS

Live Virtual Constructive Distributed Environment (LVC)

LVC Gateway,

Gateway Data Logger (GDL),

Sense and Avoid Processor (SaaProc)

Software Design Description

LVC SWDD-03

Rev D

Release Date: January 19, 2017

Software Design Description

Document No. LVC SWDD-03, Rev D

Release Date: January 19, 2016

Prepared By:

Srba Jovic – IT&E/SW and System Integration Lead, SAIC/NASA ARC

Concur:

Jamie Willhite – IT&E – LVC Integration and Test, NASA AFRC

Jim Murphy – IT&E Co-Technical Lead, NASA ARC

Sam Kim – IT&E Co-Technical Lead, NASA AFRC

Approve:

Heather Maliska – SPM, NASA AFRC

Revision Log

| Version # | Date | Change # | Page # | Author | Description |
|------------------|--------------|-----------------|---------------|---------------|--|
| Baseline | May 1, 2013 | 0 | All | Srba Jovic | Initial Release of Document |
| Rev A | Feb 19, 2014 | 1 | All | Srba Jovic | Revision to support the IHITL development phase. |
| Rev B | Apr 20, 2015 | 2 | All | Srba Jovic | Revision to support the FT3 development phase. |
| Rev C | Feb 19, 2016 | 3 | All | Srba Jovic | Revision to support the FT4 development phase. |
| Rev D | Jan 19, 2017 | 4 | All | Srba Jovic | Revision to support the ACAS Xu FT2 |

Table of Contents

| | | |
|---------|---|----|
| 1.0 | Introduction | 6 |
| 2.0 | Reference Documents..... | 7 |
| 3.0 | Software Design and Development Considerations | 9 |
| 3.1 | Programming languages..... | 9 |
| 3.2 | LVC Software | 9 |
| 3.3 | LVC Source Code Management | 9 |
| 4.0 | System Architecture Design | 10 |
| 5.0 | LVC Gateway Design..... | 12 |
| 5.1 | LVC Gateway Class Diagram..... | 13 |
| 5.2 | LVC System Data Flow | 15 |
| 5.2.1 | AFRC LVC System | 15 |
| 5.2.1.1 | AFRC LVC System Data Flow..... | 15 |
| 5.2.1.2 | AFRC Sequence Diagram..... | 16 |
| 5.2.2 | ARC LVC System..... | 18 |
| 5.2.2.1 | ARC LVC System Data Flow | 18 |
| 5.2.2.2 | ARC Sequence Diagram | 20 |
| 6.0 | LVC Gateway Data Logger Design | 23 |
| 7.0 | Sense and Avoid Processor (SaaProc) Design | 24 |
| 7.1 | SaaProc Class Diagram..... | 24 |
| 7.2 | SaaProc Class Diagram..... | 26 |
| 8.0 | Software Requirements Traceability and Implementation | 27 |
| 9.0 | Appendix A – Acronyms | 28 |

List of Figures

| | |
|--|----|
| Figure 1. ACAS Xu FT2 System Configuration Diagram..... | 8 |
| Figure 2. LVC System Architecture for the ACAS Xu FT2 Configuration running at NASA Armstrong. | 10 |
| Figure 3. LVC System Architecture for the ACAS Xu FT2 Configuration running at NASA Ames | 11 |
| Figure 4. UML Class Diagram of LVC Gateway | 14 |
| Figure 5. Data Flow Diagram for the System Configuration at NASA Armstrong: CPDS/IOServer provides relevant live data in real time | 15 |
| Figure 6. UML Sequence Diagram: Conflict Detection using ACAS Xu nominal alerting progression. | 18 |
| Figure 7. Data Flow Diagram for the System Configuration at NASA Ames: CPDS/IOServer provides relevant live data in real time | 19 |
| Figure 8. UML Sequence Diagram: Conflict Detection using ACAS Xu nominal alerting progression. | 22 |
| Figure 9. UML Class Diagram of LVC Gateway Data Logger | 23 |
| Figure 10. UML Class Diagram of SaaProc | 25 |
| Figure 11. UML Sequence Diagram for corrective TCAS RA in SaaProc | 26 |

1.0 Introduction

The desire for a capability to fly government and commercial Unmanned Aircraft Systems (UAS) in the National Airspace System (NAS) is of increasing urgency. The application of unmanned aircraft to perform national security, defense, scientific, and emergency management functions are driving the critical need for less restrictive access by UAS to the NAS. UAS represent a new capability that will provide a variety of services in the government (public) and commercial (civil) aviation sectors. The growth of this potential industry has not yet been realized due to the lack of a common understanding of what is required to safely operate UAS in the NAS.

NASA's UAS Integration into the NAS Project is conducting research in the areas of Detect and Avoid/Separation Assurance Interoperability (DAA), Human Systems Integration, Communication, and Certification to support reducing the barriers of UAS access to the NAS. To accomplish this task, the UAS-NAS Project will conduct a series of Integrated Human-in-the-Loop (IHITL) and Flight Test activities that integrate key concepts, technologies and/or procedures in a relevant air traffic environment. Each of the integrated events will build on the technical achievements, fidelity and complexity of the previous tests and technical simulations, resulting in a body of evidence that supports the development of regulations governing the access of UAS into the NAS.

The IHITL Test configuration was described in Revision A of this document. The Flight Test 3 (FT3) test configurations were described in Revision B of this document. Flight Test 4 (FT4) was described in Revision C of this document. ACAS Xu Flight Test 2 (ACAS Xu FT2) is described in Revision D of this document.

Figure 1 provides a high-level diagram of the ACAS Xu FT2 System Configuration. The System Configuration is designed to support pairwise one/two-intruder encounters of the low speed Ownship (Ikhana Predator MQ9) with Intruder/s across the entire set of test cards. The aircraft will be flown in the Armstrong Flight Research Center (AFRC) test range. LVC subsystems such as Conflict Prediction and Display System (CPDS), Vigilant Spirit Control Station (VSCS), SaaProc/JADEM, and the Thales ADS-B track data source will interface with LVC Gateway to relay critical research data, which will be recorded by the LVC Data Logger and other system recording devices for the post-run data processing and data analysis phase. VSCS, and SaaProc/JADEM are not required to meet FT2 minimum success, but will be used in a shadow mode as a risk reduction for upcoming tests.

The CPDS in conjunction with the I/O Server function as the interface for receiving relevant live Ownship and Intruder track data, and ACAS Xu alerting data, which are subsequently published to the LVC environment as shown in Figure 1 and Figure 2. The Vigilant Spirit Control Station (VSCS) will be used to present ACAS Xu alerts and guidance (both vertical and horizontal) to the VSCS display. SaaProc/JADEM will also calculate and record DAA alerts and guidance parameters based on the state data for the Ownship and Intruders and publish those messages to the LVC Gateway. The LVC Data Logger subscribes to all messages that are published to the LVC Gateway and will record and store them in csv data file format. Note that VSCS, SaaProc/JADEM and ACAS Xu applications will run only in the shadow mode at NASA Ames (labeled as Remote LVC in Figure 1).

LVC system clients rely on the LVC Gateway to exchange shared information with each other. LVC Gateway and LVC Data Logger represent the core components of the ACAS Xu FT2 LVC distributed test environment.

This document provides the software design description for the core software components (LVC Gateway and the LVC Gateway Data Logger) and the SAA Processor (SaaProc) that the IT&E team is responsible for.

Note that this SWDD traces implementation to the LVC SWRD-02 Rev E requirements for ACAS Xu FT2 (See Section 8).

2.0 Reference Documents

| Document Title | Description |
|-------------------------------------|--|
| APR-7150.2 | <i>Ames Software Engineering Requirements</i> |
| IT&E-CMP-001 | <i>IT&E Configuration Management Plan</i> |
| IT&E PP-01 | <i>IT&E Project Plan</i> |
| IT&E RMP-01 | <i>UAS-NAS Risk Management Plan</i> |
| IT&E SAP-01 | <i>IT&E Software Assurance Plan</i> |
| UAS-NAS.ITE.FT4. ORD.20160122.R0 | <i>Objectives and Requirements Document (ORD)</i> |
| LVC SRD-01 | <i>System Requirements Document</i> |
| LVC-SWRD-02 | <i>Software Requirements Document</i> |
| IT&E SDMP-01 | <i>Software Development & Management Plan</i> |
| LVC V&VP-01 | <i>LVC Verification & Validation Test Plan</i> |
| LVC Gateway ICD-03 | <i>LVC Gateway Message Interface Control Documents</i> |

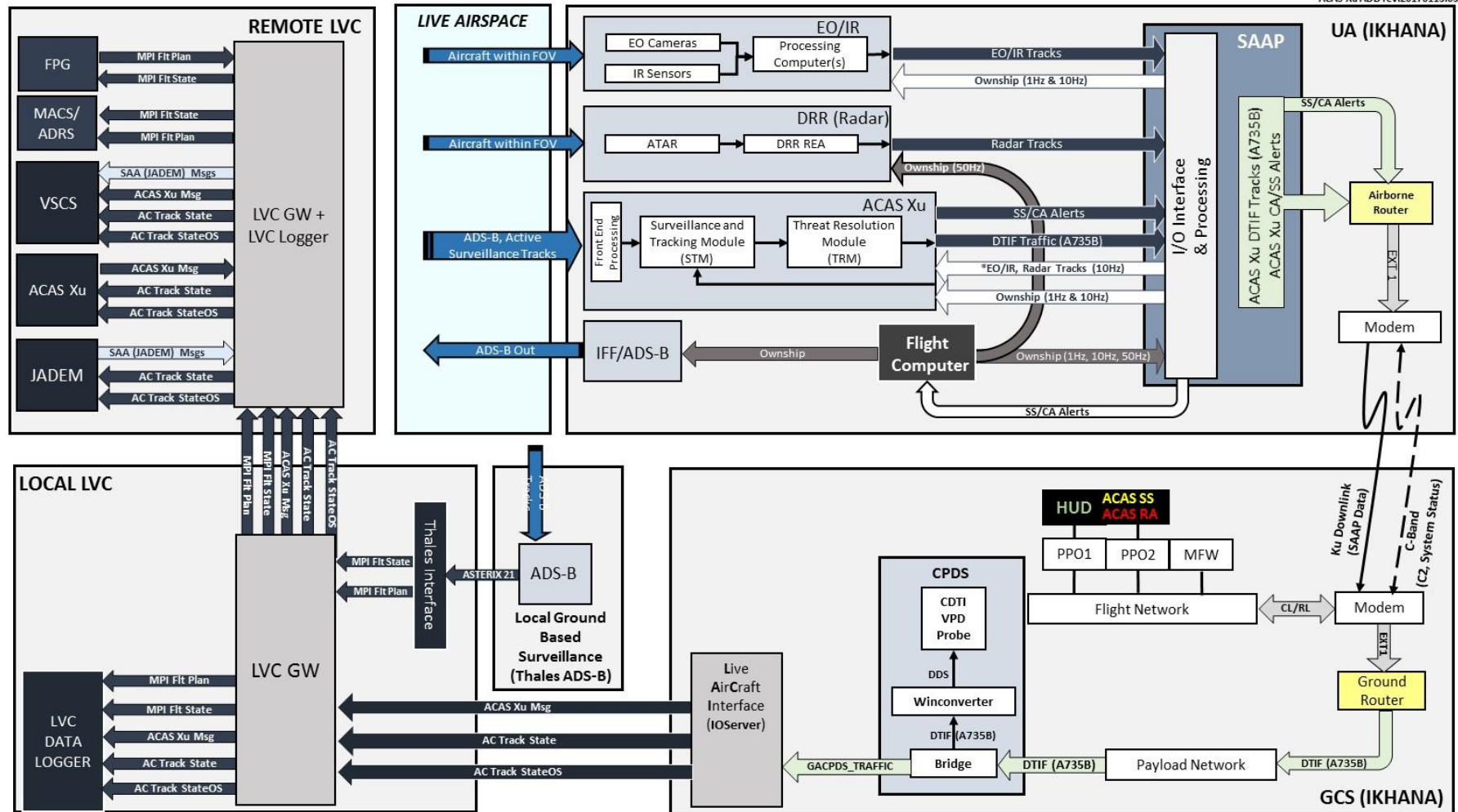


Figure 1. ACAS Xu FT2 System Configuration Diagram

3.0 Software Design and Development Considerations

3.1 Programming languages

The LVC Gateway software development uses the C++ programming language and Object Oriented Design methodology enabled by the language paradigm. The Object Oriented abstraction concept will be utilized for the network communication classes to capture specifics of different clients' interfaces as well as for configuring socket types as a TCP client or server. The design can accommodate other protocols such as UDP, Multicast or Loopback sockets. Some of the hardware interfaces and libraries are best integrated using the C++ language due to the timeframe in which these libraries were created.

3.2 LVC Software

The LVC is a complex system designed and developed by the combined NASA/ARC and NASA/AFRC IT&E teams to integrate with other Government off-the-shelf (GOTS) software. At NASA Ames, software development is guided by the Ames LVC Software Development Plan as updated for ACAS Xu. This includes the typical Configuration Change Board (CCB) and Configuration Management (CM) processes followed by SimLabs at ARC. Integration and test at NASA AFRC are per the IT&E configuration management process as defined in the IT&E CM Plan.

3.3 LVC Source Code Management

Version control for the LVC software is maintained using the Concurrent Versions System (CVS). Each revision of the source code will be checked into the source code configuration management repository using CVS, providing complete traceability of all software changes for each application build and executable.

The LVC Gateway, LVC Gateway Data Logger, and SaaProc software have been determined to be class D software as defined in APR 7150.2.

4.0 System Architecture Design

The LVC System Architecture for the ACAS Xu FT2 is shown in Figure 2 and Figure 3. The LVC Gateway acts to integrate participating components.

NASA Armstrong system configuration is shown in Figure 2. Note that the LVC Gateway, and LVC Data Logger have been modified to handle connectivity requirements and messaging associated with CPDS/IOServer per ACAS Xu FT2 requirements.

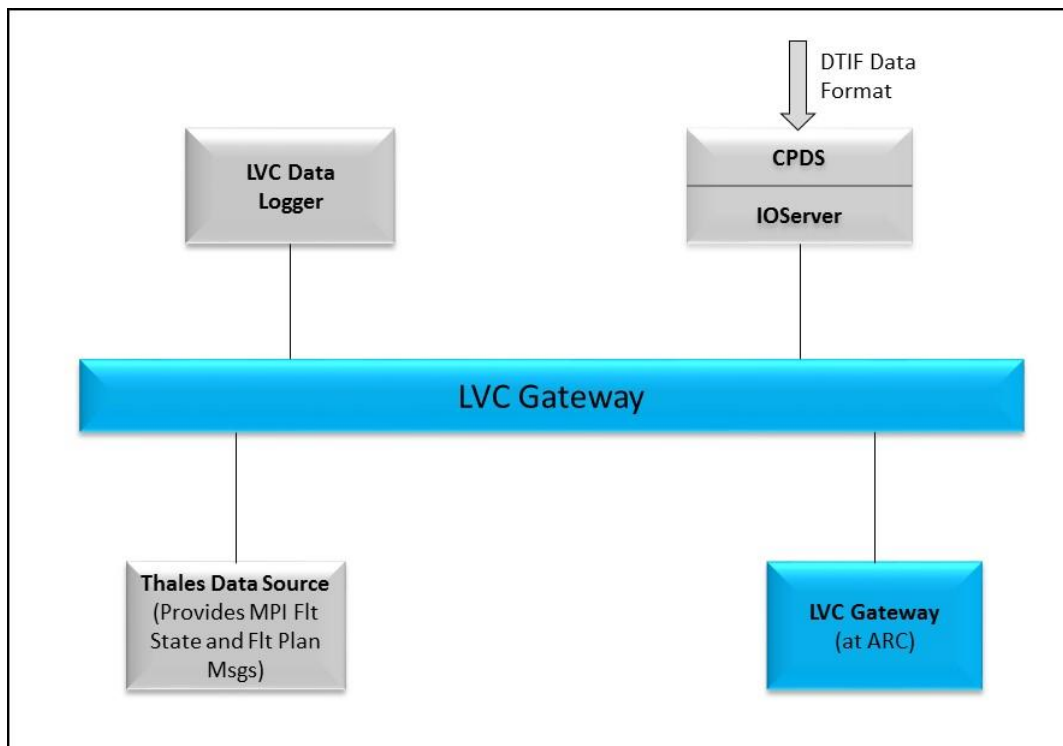


Figure 2. LVC System Architecture for the ACAS Xu FT2 Configuration running at NASA Armstrong.

NASA Ames system configuration is shown in Figure 3. Note that the Ames configuration will be operating in the shadow mode.

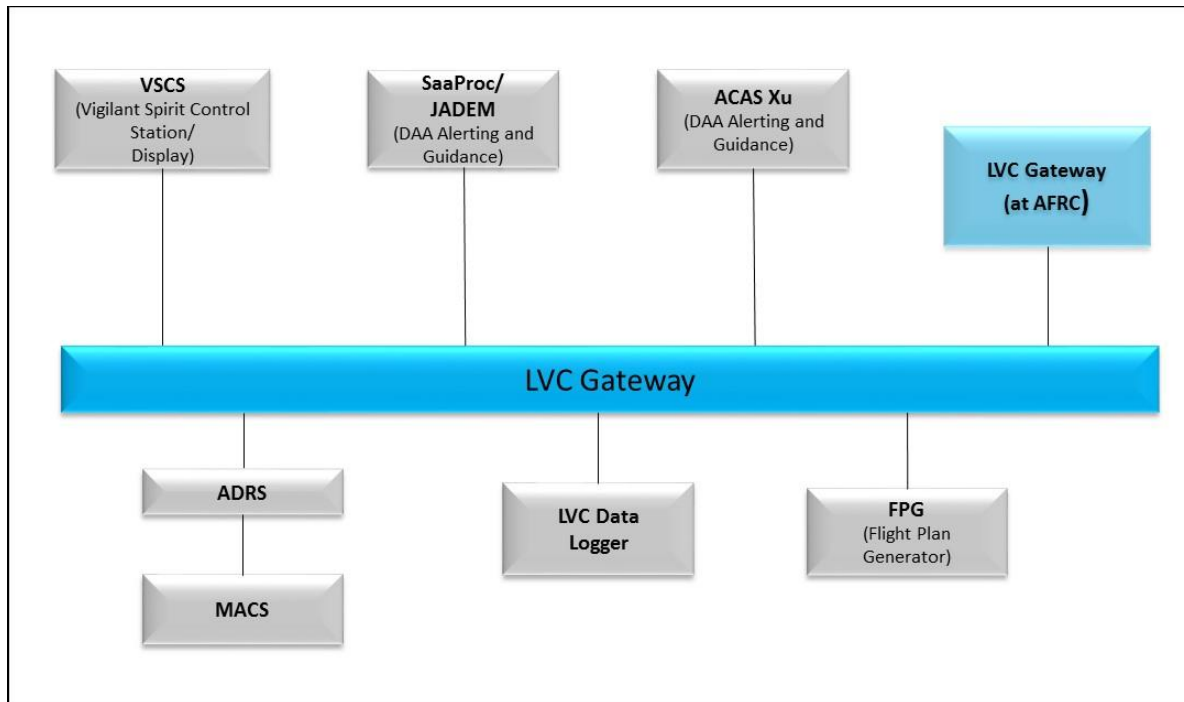


Figure 3. LVC System Architecture for the ACAS Xu FT2 Configuration running at NASA Ames

5.0 LVC Gateway Design

The LVC Gateway and the LVC Data Logger constitute the core components of the LVC system for ACAS Xu FT2. The participating applications connect as clients to the LVC Gateway's socket server. The LVC Gateway ICD-03 Rev-F provides data structures definitions in support of the project requirements. In addition to the data structures that supported the FT4 and Part Task 6 experiments, the ICD has been updated to include the ACAS Xu alerting and guidance message, and two new DAA type track state messages (Aircraft Track State for both Ownship and Intruders) compliant to the current SC228 DAA MOPS. Upon initial network connection, each client will send a handshake message to the Gateway defining the message types the client will publish and subscribe.

The ICD specifies details for data exchange between the LVC Gateway and the following clients:

- a) Running at NASA Armstrong:
 - LVC Data Logger
 - Thales ADS-B Receiver
 - CPDS/IOServer
- b) Running at NASA Ames in the shadow mode
 - Vigilant Spirit Control Station (VSCS)
 - SaaProc/JADEM
 - ACAS Xu
 - LVC Data Logger
 - Multi Aircraft Control System (MACS)

All clients listed above will connect to the LVC Gateway using the common client port interface of the LVC Gateway server's TCP socket. In addition to the legacy MPI set of messages used in the previous UAS-in-the-NAS experiments, the clients will be required to implement three new messages - ACAS Xu Message, and Aircraft Track State for Ownship and for Intruder.

Also, the LVC Gateway will send a periodic heartbeat message to each client to check the connection health. If the socket connection does not reply in a timely manner due to a client process crash, process disconnect (voluntary or involuntary) or socket failure, the LVC Gateway will send the delete message out to the LVC system for all tracks generated by that client.

5.1 LVC Gateway Class Diagram

The Unified Modeling Language (UML) class diagram of the LVC Gateway design is shown in Figure 4. The figure depicts the software architecture representing the relationship between the classes of the LVC Gateway.

The Extensible Markup Language (XML) input configuration file is used to define the LVC Gateway configuration in terms of interface types, interface socket types, and data message types that are transmitted by the LVC Gateway.

The GW Client Manager class is responsible for the instantiation of all LVC Gateway components in the initialization phase of the code based on the XML configuration file. The GW Client Manager contains a collection of GW Client objects. In addition, it utilizes the observer/observable pattern. Using this paradigm, the *observable* is associated with clients that publish messages defined in the ICD while the *observer* is associated with clients that subscribe to those messages.

The MPI, ADRS, LARC specialized interface classes are derived from the abstract GW Client class as defined by the XML configuration file. Each GW Client can have many TCP and/or UDP sockets of the client and/or the server type. In addition to the data structures that supported FT4 experiment, each interface on the LVC Gateway side will receive three new messages that are added for ACAS XU FT2, i.e., 1) ACAS Xu message (MsgAcasXu) containing DAA vertical and horizontal alerts and guidance message, 2) Aircraft Track State for the Ownship (MsgAcTrackStateOS), and 3) Aircraft Track State for Intruders (MsgAcTrackState). All messages are time stamped, and then passed to the subscribing clients. There are three time stamp data field members in the new track state messages – the time of applicability, time when the message was sent, and time received at the LVC Gateway. This multiplicity of time stamps will allow better differentiation and reporting of time latencies between data publishers and receivers.

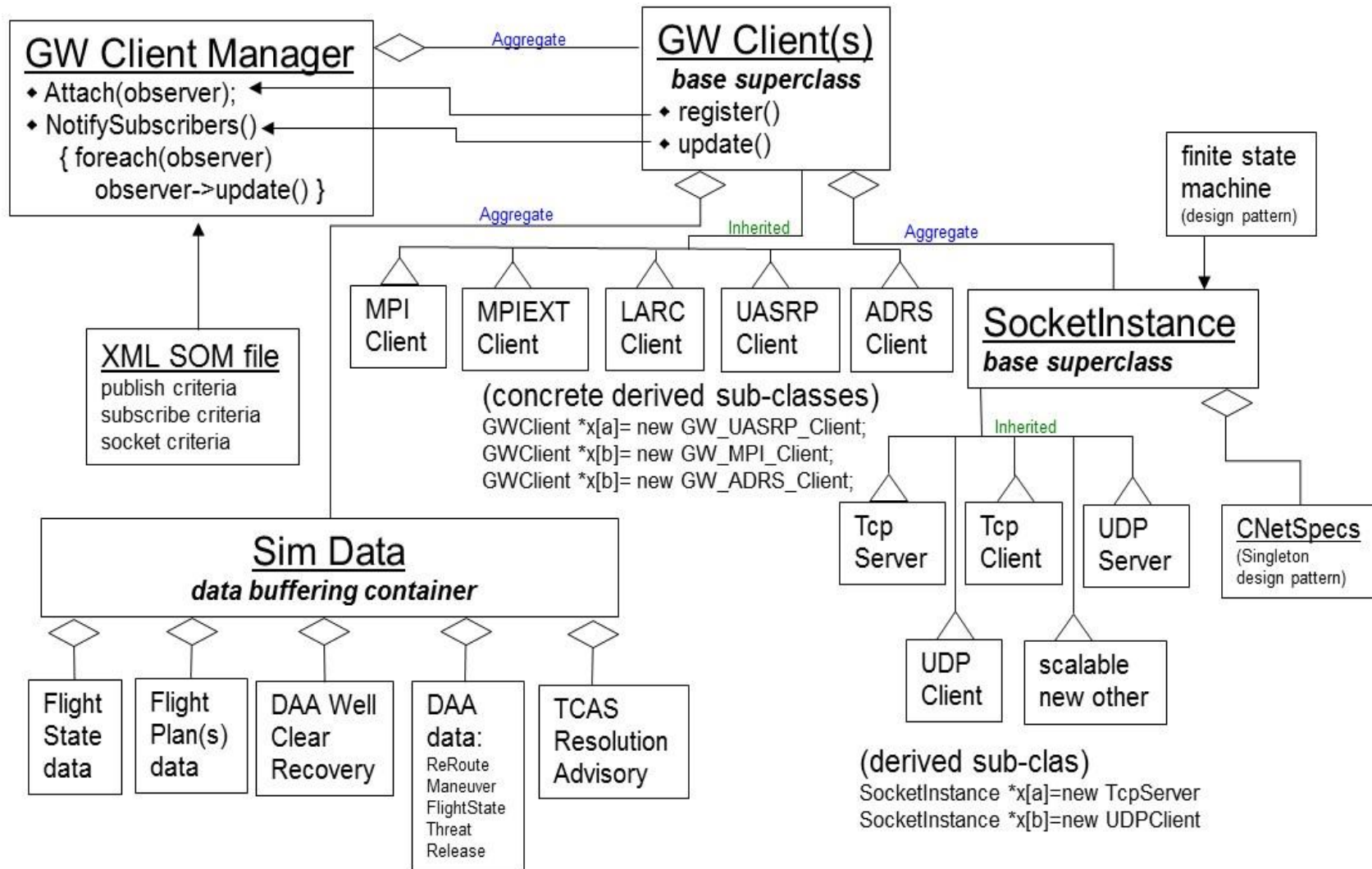


Figure 4. UML Class Diagram of LVC Gateway

5.2 LVC System Data Flow

The LVC Gateway will exchange the new and the legacy messages with LVC participants. The flow of messages between the LVC participants will be presented for the two-system configurations one running at NASA Armstrong and one running in the shadow mode at NASA Ames.

5.2.1 AFRC LVC System

The Data Flow and Sequence Diagram for the NASA AFRC system configuration are shown in subsequent sections.

5.2.1.1 AFRC LVC System Data Flow

The flow of messages between the LVC participants is depicted in Figure 5 for the LVC system configuration at NASA Armstrong.

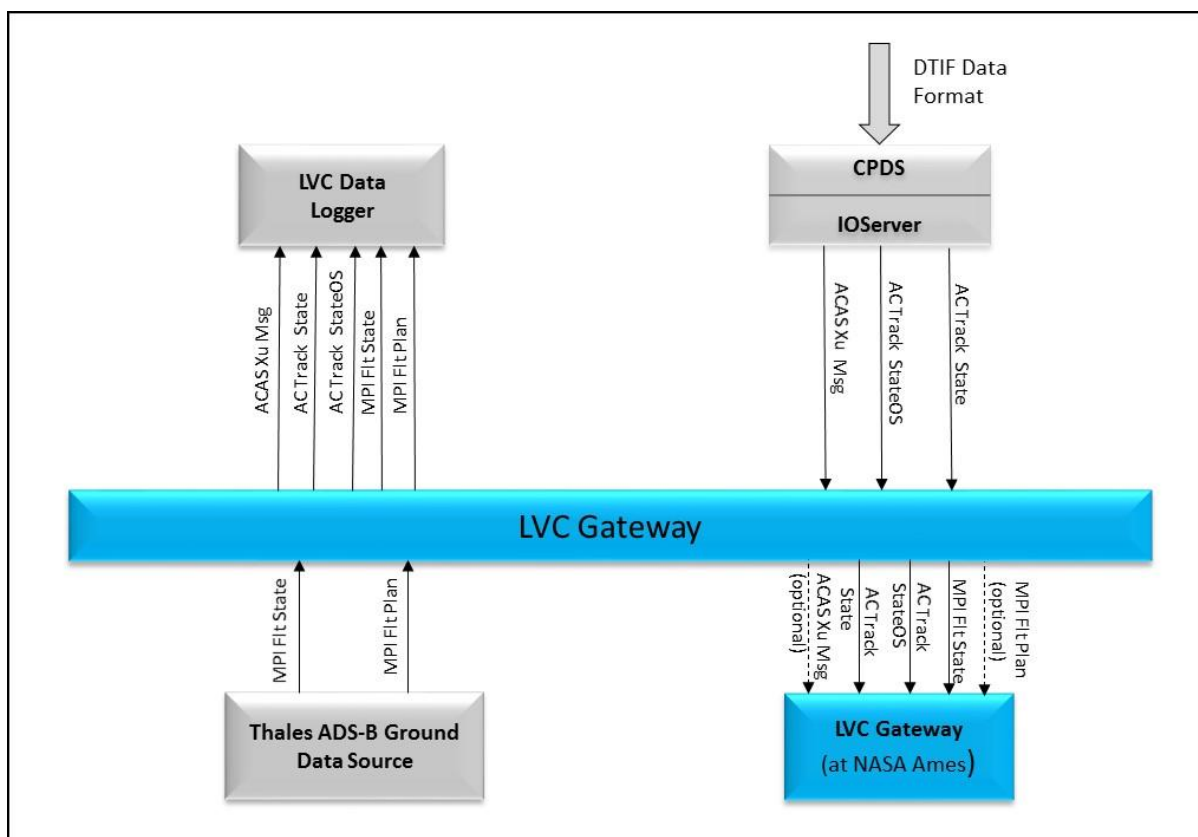


Figure 5. Data Flow Diagram for the LVC System Configuration at NASA Armstrong:
CPDS/IO Server provides relevant live data in real time

State data for live traffic is generated by the Ownship on-board surveillance sensors. The surveillance data is downlinked to the ground control station (GCS) utilizing A735B protocol and packaged into Digital Table Interchange Format (DTIF) data records at a 1Hz data rate. CPDS/IOserver copies records for each target as well as ACAS Xu records from the GA-ASI Live domain and maps them into the corresponding LVC data structures conforming to the LVC ICD-03. Subsequently, the data structures are sent to the LVC Gateway as depicted in Figure 5. The IOserver is responsible for mapping the three new message types from CPDS into the data packets conforming to the LVC ICD-03.

Note: It has been determined by the stakeholders that Intruder tracks will be passed through the LVC without any range filtering (with respect to Ownship), and each application will be responsible to handle and filter Intruder tracks based on its own display requirements. The received Intruders in previous software versions were a subset of the Ownship's background traffic as determined by the Ownship's sensor range filter via the SAA Flight State message (see previous SWDD-3 Rev C for FT4).

Thales ADS-B ground based surveillance source generates additional surveillance data. The Thales wrapper maps the ADS-B data type into the MPI Flight State. The wrapper will generate MPI Flight Plan for each target in the ADS-B surveillance range.

The LVC Data Logger records the following messages: ACAS-Xu, Aircraft Track State for the Ownship and Intruders, MPI Flight State updated at 1Hz data rate, and MPI Flight Plan.

5.2.1.2 AFRC Sequence Diagram

The UML sequence diagram associated with the message flow between the LVC system components are shown in Figure 6 for selected use case. The sequence diagram is shown for the received live traffic from IOserver when an encounter between the Ownship and an Intruder occurs. An encounter will produce an ACAS Xu alerting and guidance message. The tracks for the two aircraft and the alerting and guidance data are transmitted via LVC Gateway to LVC Data Logger.

A list of sequence of events is provided below:

1. The surveillance system of the live aircraft (Ikhana) will send the state data update of the Ownship (Ikhana) and Intruders that are in the range of the on-board sensors to the Ikhana GCS at the rate of 1Hz.
2. The CPDS/IOserver will receive the state data messages of the live traffic via the DTIF data formatted record. Subsequently, IOserver will map the corresponding state data fields into the Aircraft Track State for the Ownship (ACTrackStateOS) and Aircraft Track State for Intruders (ACTrackState) message data structures.
3. The CPDS/IOserver will receive ACAS Xu alerts and guidance data via the DTIF data

formatted record. Subsequently, IOServer will map the received corresponding data fields into the ACAS Xu message structure as defined in LVC ICD.

4. The CPDS/IOServer will send two track state messages (ACTrackStateOS and ACTrackState) to the LVC Gateway.
5. The LVC Gateway receives track state messages (ACTrackStateOS and ACTrackState) at the nominal rate of 1Hz.
 - a. LVC Gateway publishes track state messages (ACTrackStateOS and ACTrackState) to LVC Data Logger at the nominal data rate of 1Hz.
 - i. LVC Data Logger receives both tracks.
 - ii. LVC Data Logger logs both tracks in the csv data file. Note that LVC Data Logger generates a csv data file at the beginning of each test.
6. The CPDS/IOServer will publish the ACAS Xu message to the LVC Gateway.
7. The LVC Gateway receives ACAS Xu message.
 - a. LVC Gateway publishes ACAS Xu message to LVC Data Logger.
 - i. LVC Data Logger receives ACAS Xu message.
 - ii. LVC Data Logger logs ACAS Xu message in the csv data file.
8. The Thales Interface receives ADS-B message from the external surveillance station and maps it into MPI Flight State message.
9. The Thales Interface generates MPI Flight Plan message for every new target that is in the range of the ADS-B. Note that only one copy of MPI Flight Plan message is generated for the given target.
10. The Thales Interface publishes MPI Flight State to LVC Gateway.
 - a. LVC Gateway receives MPI Flight State message.
 - b. LVC Gateway publishes MPI Flight State message to LVC Data Logger.
 - i. LVC Data Logger receives MPI Flight State message.
 - ii. LVC Data Logger logs MPI Flight State message to the csv data file.
11. The Thales Interface publishes MPI Flight Plan to LVC Gateway.
 - a. LVC Gateway receives MPI Flight Plan message.

- b. LVC Gateway publishes MPI Flight Plan message to LVC Data Logger.
 - i. LVC Data Logger receives MPI Flight Plan message.
 - ii. LVC Data Logger logs MPI Flight Plan message to the csv data file.
- 12. The process repeats from Step 1 when the new traffic update arrives into IOServer.

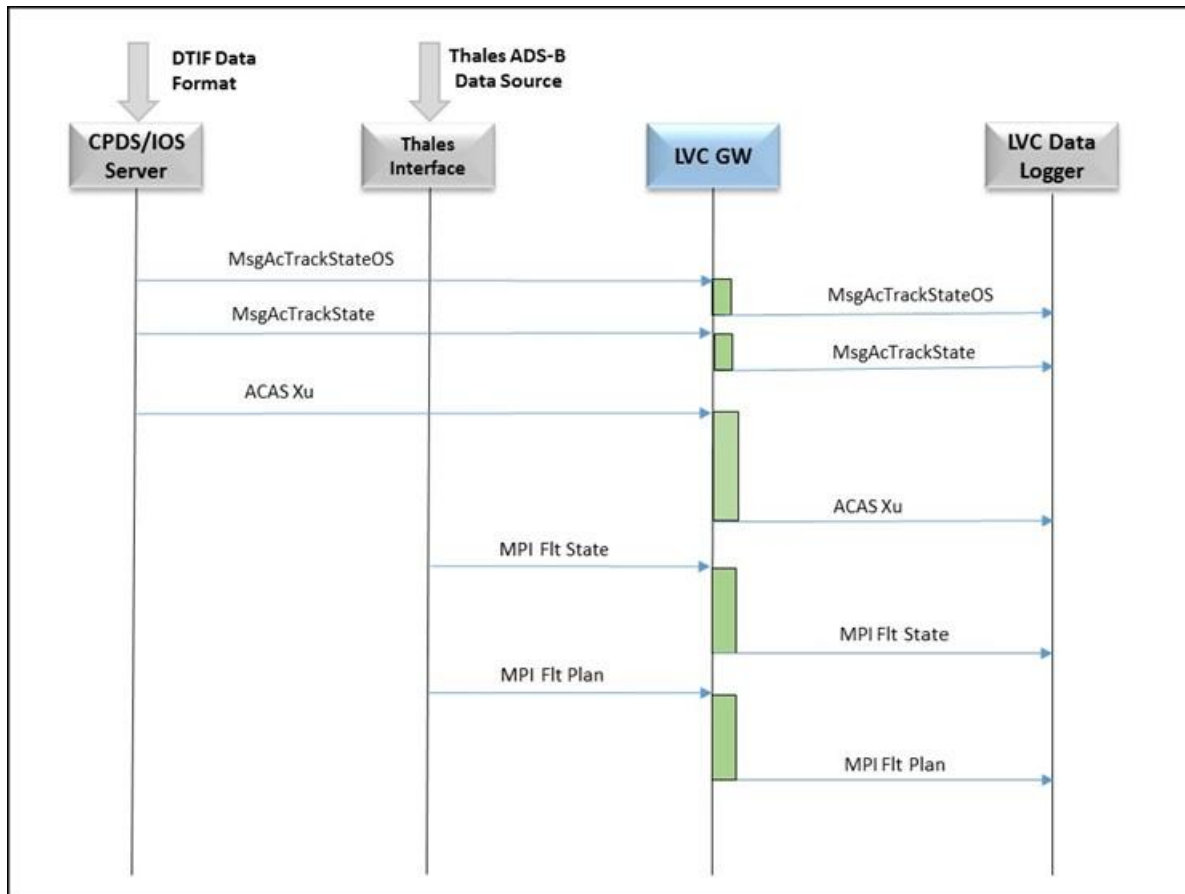


Figure 6. UML Sequence Diagram for the LVC System Configuration at NASA Armstrong: Conflict Detection using ACAS Xu nominal alerting progression.

5.2.2 ARC LVC System

The Data Flow and Sequence Diagram for the NASA Ames system configuration is shown in subsequent sections.

5.2.2.1 ARC LVC System Data Flow

The flow of messages between the LVC participants is depicted in Figure 7 for the LVC system configuration at NASA Ames.

State data for live traffic is generated by the Ownship on-board surveillance sensors. The surveillance data is downlinked to the ground control station (GCS) utilizing A735B protocol and packaged into Digital Table Interchange Format (DTIF) data records at a 1Hz data rate. CPDS/IOserver copies records for each target as well as ACAS Xu records from the GA-ASI Live domain and maps them into the corresponding LVC data structures conforming to the LVC ICD. Subsequently, the data structures are sent to the LVC Gateway as depicted in Figure 7. The IOserver is responsible for mapping the three new message types from CPDS into the data packets conforming to the LVC ICD.

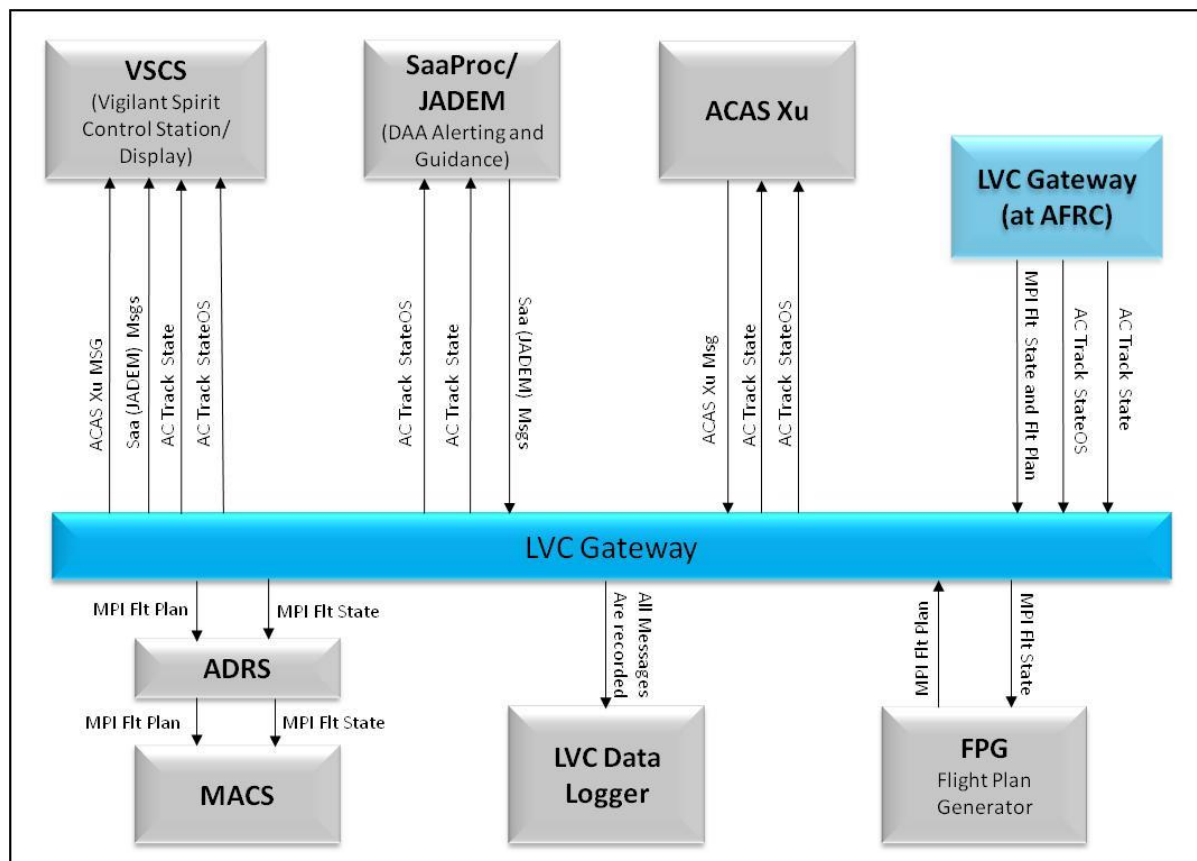


Figure 7. Data Flow Diagram for the LVC System Configuration at NASA Ames:
CPDS/IOserver provides relevant live data in real time

The VSCS will receive and display the Aircraft Track State for Ownship and Intruders, ACAS Xu, and JADEM set of messages.

Note: It has been determined by the stakeholders that Intruder tracks will be passed through the LVC without any range filtering (with respect to Ownship), and each application will be responsible to handle and filter Intruder tracks based on its own requirements. The received Intruders in previous software versions were a subset of the Ownship's background traffic as determined by the Ownship's sensor range filter via the SAA Flight State message (see previous

SWDD-3 Rev C for FT4).

The JADEM input is comprised of the following messages: Aircraft Track State for the Ownship and Intruders updated at 1Hz data rate, and the Ownship Navigation Mode. The DAA algorithm calculates the following set of output DAA messages for Intruders: Threat results (Alerts), Resolution Maneuvers, Resolution Reroute and Well Clear Recovery (WCR). All messages are defined in the LVC ICD-03.

5.2.2.2 ARC Sequence Diagram

The UML sequence diagram associated with the message flow between the LVC systems components is shown in Figure 8 for the selected use case. The sequence diagrams are shown for the received live traffic from IOServer when an encounter between the Ownship and an Intruder occurs. An encounter will produce an ACAS Xu alerting and guidance message. The tracks for the two aircraft and the alerting and guidance data are transmitted via LVC Gateway to VSCS for the display. In addition, JADEM alerts and guidance messages are calculated for the same pair of aircraft and are published to the LVC Gateway. The data is also sent to the VSCS for display.

A list of sequence of events is provided below:

1. The LVC Gateway running at NASA Armstrong (AFRC) publishes Aircraft Track State for the Ownship (ACTrackStateOS), Aircraft Track State for Intruders (ACTrackState), and MPI Flight State to the LVC Gateway at NASA Ames at the nominal rate of 1Hz.
2. The LVC Gateway running at Ames receives the two track state messages.
 - a. LVC Gateway publishes track state messages (ACTrackStateOS and ACTrackState) to LVC Data Logger at the nominal rate of 1Hz.
 - i. LVC Data Logger receives the two state messages and logs them in a data file.
 - b. LVC Gateway publishes track state messages (ACTrackStateOS and ACTrackState) to VSCS at the nominal rate of 1Hz.
 - i. VSCS displays both tracks in the VSCS display. Note that only a subset of Intruders will be displayed as the Intruders will be subjected to the sensors range model, which is an integral part of VSCS. This capability will be configurable to either pass through all the tracks or to subject each track to the ADS-B and RADAR range filter.
 - c. LVC Gateway publishes track state messages (ACTrackStateOS and ACTrackState) to SaaProc/JADEM at the nominal rate of 1Hz.
 - i. As JADEM receives the track data, the sensors range model, which is an

integral part of the JADEM, will be invoked, when a set of Intruders is determined based on their 3D spatial distance from the Ownship. This capability can be configured to either pass through all the tracks or to subject each track to the ADS-B and RADAR range filter and pass only a subset of the entire traffic to the JADEM algorithm. Intruders are subset of the background traffic passed from the Ownship to the GCS.

1. DAA alerting algorithm in JADEM calculates threat level and guidance based on received ACTrackStateOS and ACTrackState.
 2. SaaProc/JADEM publishes JADEM alert and guidance message to LVC Gateway.
- ii. LVC Gateway receives JADEM alerts and guidance message and sends it to VSCS.
- iii. LVC Gateway publishes JADEM alerts and guidance message to VSCS.
 1. VSCS receives JADEM alert and guidance message and displays only the vertical component of the message. Note that VSCS might use JADEM vertical alert and guidance to supplement ACAS Xu messages as ACAS Xu is currently not supporting vertical alerting and guidance prediction.
 2. VSCS might use JADEM vertical alerts and guidance to supplement ACAS Xu messages as ACAS Xu is currently not supporting vertical alerting and guidance prediction.
- d. LVC Gateway publishes MsgAcTrackStateOS and MsgAcTrackState messages to ACAS Xu application at the nominal rate of 1Hz.
 - i. ACAS Xu receives the two track state messages.
 1. ACAS Xu calculates alerts and advisories.
 2. ACAS Xu publishes ACAS Xu message to LVC Gateway.
 - ii. LVC Gateway receives ACAS Xu message.
 - iii. LVC Gateway publishes ACAS Xu message to VSCS.
 1. VSCS receives and displays ACAS Xu alerts and advisories.
3. LVC Gateway at Armstrong publishes MPI Flight State message to LVC Gateway at Ames at the nominal rate of 1Hz.

4. LVC Gateway at Ames receives MPI Flight State message.
 - a. LVC Gateway publishes MPI Flight State message to LVC Data Logger at the nominal rate of 1Hz.
 - i. LVC Data Logger receives and logs it in a data file created at the beginning of the run.
5. The process repeats from Step 1 when the new traffic update arrives into IOServer.

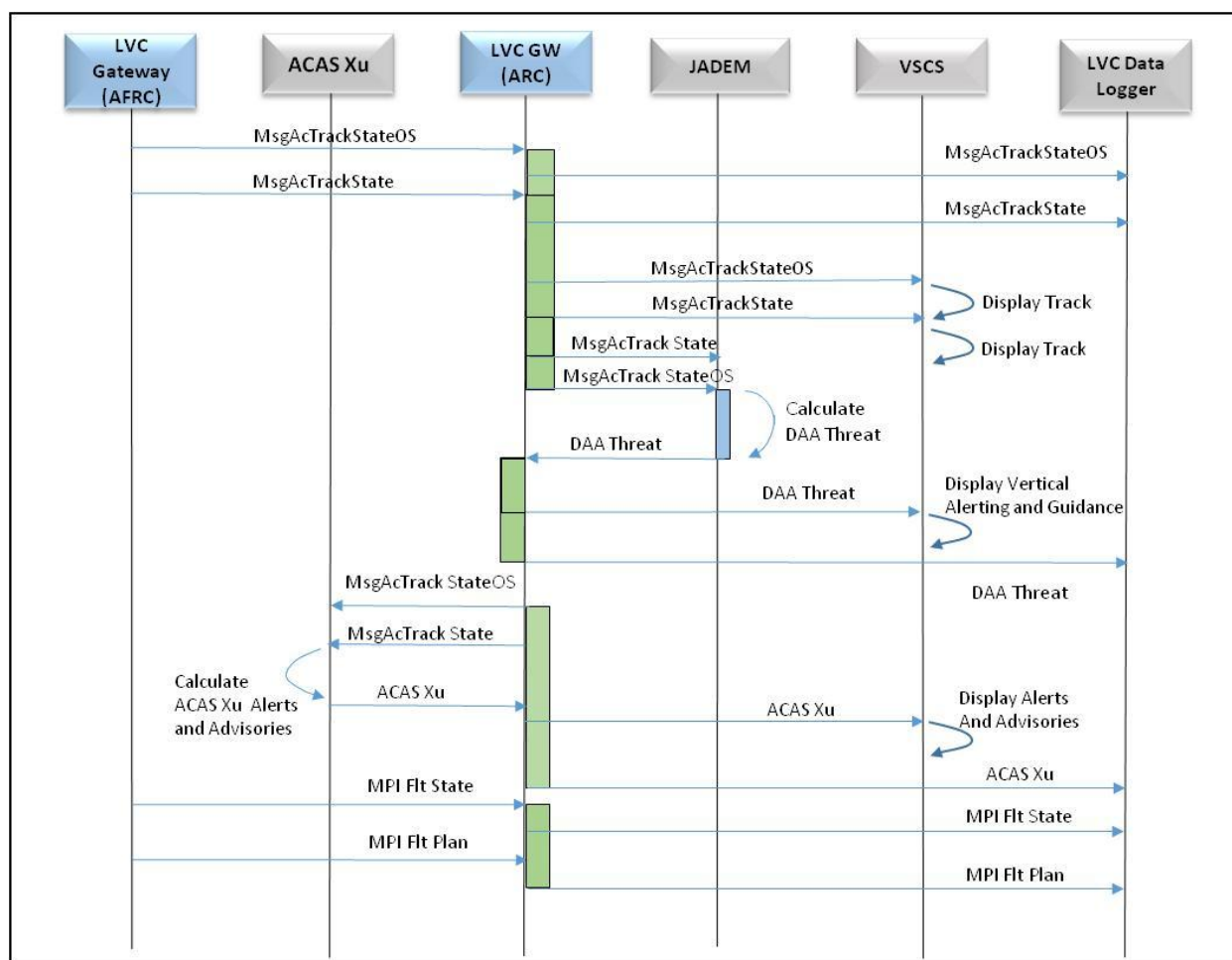


Figure 8. UML Sequence Diagram for the LVC System Configuration at NASA Ames: Conflict Detection using ACAS Xu nominal alerting progression.

6.0 LVC Gateway Data Logger Design

The LVC Gateway Data Logger (GDL) is designed as a client to the LVC Gateway using TCP/IP client socket interface protocol. The GDL interface conforms to the LVC Gateway ICD. The handshake message will specify that the GDL will subscribe to and record all messages transmitted by the LVC Gateway. The list of messages is as follows: MsgAcTrackStateOS, MsgAcTrackState, MsgACAS-Xu, DAA Threat Results, DAA Omni Bands, DAA Well Clear Recovery, and Aircraft Delete messages. The messages are logged when the application is set for the recording mode by the command line parameter before the application is launched. Data messages of every type will be recorded in an ASCII comma-delimited csv data file. The UML class high-level diagram of the LVC Gateway Data Logger design is shown in Figure 9.

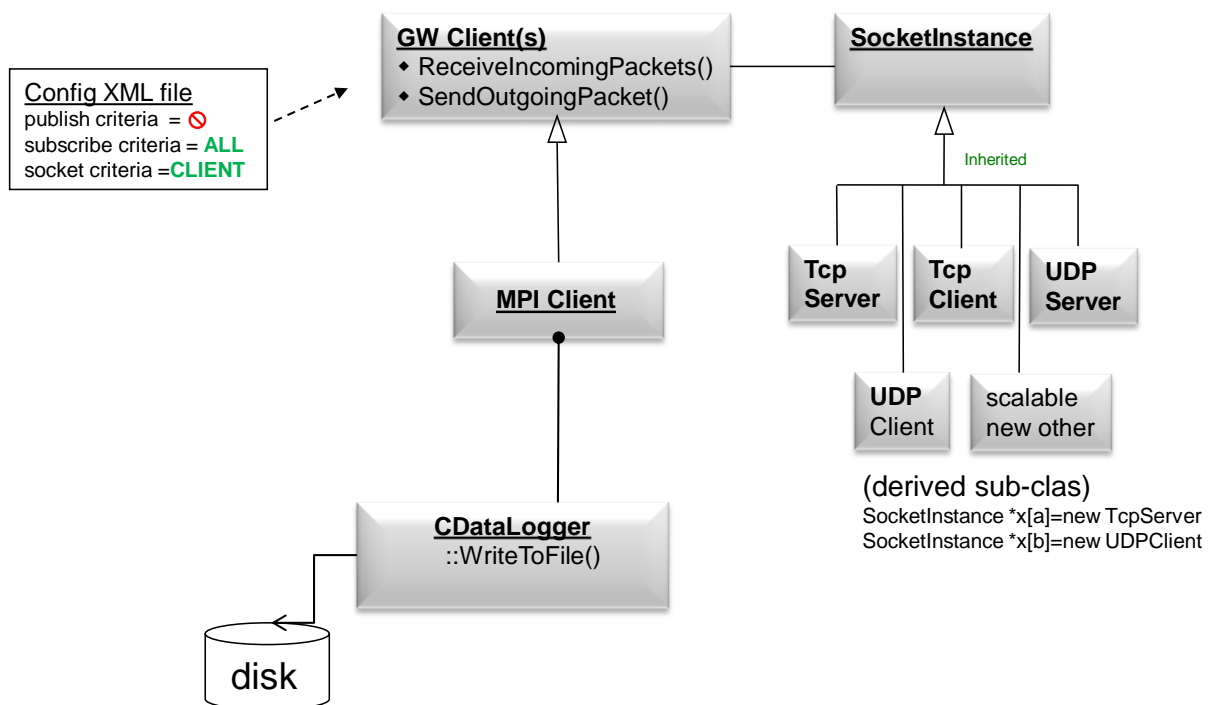


Figure 9. UML Class Diagram of LVC Gateway Data Logger

7.0 Sense and Avoid Processor (SaaProc) Design

The SaaProc subsystem is an integral component of the UAS Ground Control Station (GCS) under test, providing a framework for integrating multiple Detect and Avoid (DAA) algorithm and traffic display combinations. The SaaProc communicates over a socket connection as a client with the LVC Gateway. The internal IO data infrastructure supports the flow of inputs to the DAA algorithm such as Ownship, and Intruder flight data, and TCAS RA messages, while outputs from the DAA algorithm are DAA Threat Results, DAA Omni Bands, and DAA Well Clear Recovery messages. The SaaProc output is published to the LVC Gateway. The SaaProc class diagram is shown in Figure 10.

Threat levels and resolutions produced by DAA are displayed graphically on VSCS. The Sequence Diagram of Figure 11 shows the internal sequence infrastructure for the SaaProc subsystem.

A multithreaded infrastructure is employed to handle different data rate requirements for data management as well as handling asynchronous events associated with trial planning and JADEM's conflict detection.

An additional alerting capability is provided in the form of DAA Omni Bands. The DAA Omni Bands algorithm is implemented in the JADEM environment. The Omni Bands provide threat intervals for heading and altitude as defined in the LVC ICD.

7.1 SaaProc Class Diagram

The UML class diagram of the SaaProc design is shown in Figure 10. The figure depicts the software architecture representing the relationship between the classes of the SaaProc. Central to the SaaProc operation is the SaaIOController class, which encapsulates the DAA algorithm (DAA) implemented in the SenseAndAvoid class. The SaaIOController directs the flow of data coming in from LVC Gateway and the data going out to LVC Gateway as a result of the SAA computation. The input and output to and from the LVC Gateway are facilitated by the NetworkCom class.

Employing a bridge pattern in the input structure, SaaProc can adapt to pairing up different client input formats via the SaaInput abstract class with different data consumers via the FlightDataInputAdaptor class. Likewise, the output structure can adapt to pairing up different algorithm output formats via the SaaResultsAdaptor class with different output data protocols via the SaaOutput abstract class. This design provides flexibility for the SaaProc to handle any algorithm, given any type of input data format.

In order to maximize its performance, SaaProc utilizes three additional worker threads running asynchronously, implemented by SaaInput, SaaOutput, and SaaTrialPlanController. This allows the input/output/DAA computation to operate independently at a 1Hz nominal rate while the internal trial planner operates at a 15Hz rate. During the run time, these asynchronous objects communicate with each other via four synchronized objects implemented by the FlightDataLocker, ResultsLocker, TrialPlanDataLocker, and TrialPlanResultLocker.

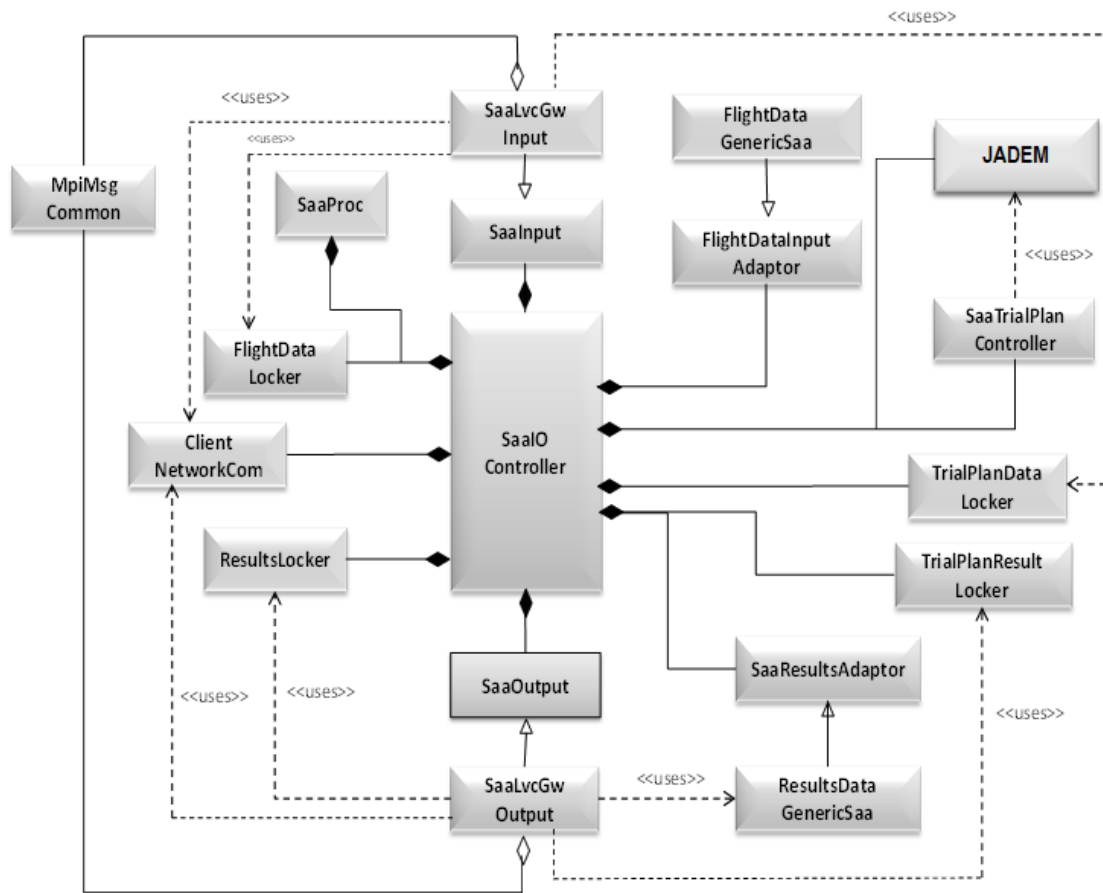
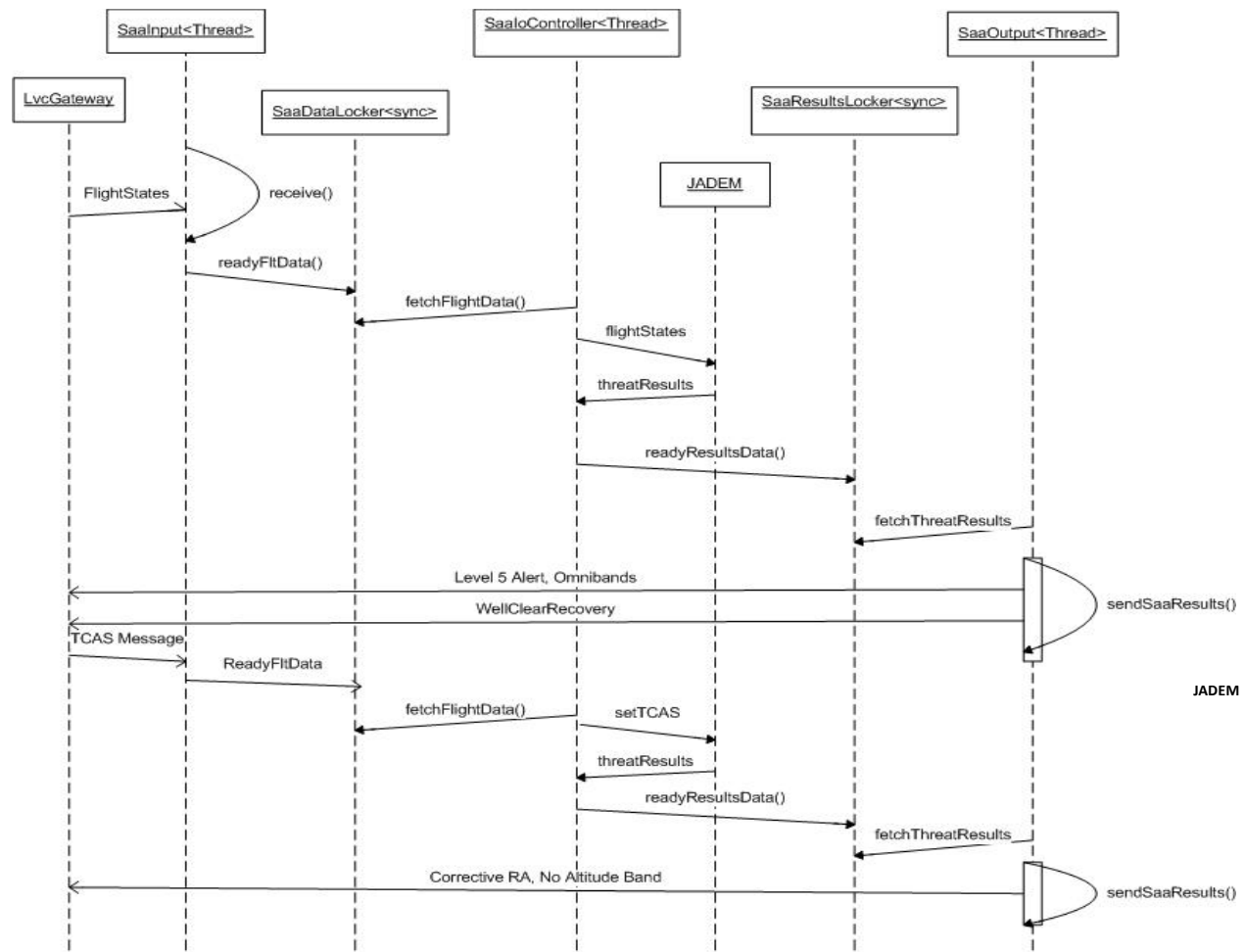


Figure 10. UML Class Diagram of SaaProc



Sequence Diagram For A TCAS Alert Scenario

Figure 11. UML Sequence Diagram for corrective TCAS RA in SaaProc

7.2 SaaProc Class Diagram

The sequence of events shows the alerting progression including WCR guidance and corrective TCAS RA:

1. The SaaInput object receives FlightState messages from the LvcGateway, which are then submitted to the JADEM component for calculation.
2. JADEM detects Level 5 threats. It computes the Omnibands for avoidance advisory.
3. SaaOutput object sends the threat results to the LvcGateway, containing the threat information along with the Omnibands.

4. As events evolve, TCAS alert is generated by the TCAS server which is then forwarded to the LVC Gateway.
5. SaaOutput object receives the TCAS message and submits it to the JADEM component for calculation.
6. JADEM takes the TCAS parameters into consideration and re-computes the threat results.
7. In this particular scenario, JADEM generates a Corrective RA and inhibits the Altitude Bands.
8. SaaOutput object sends the results from Step 7 above to the LVC Gateway, to be consumed by clients such as the VSCS.

Note that VSCS incorporates integrated advanced display capabilities in support of the Detect and Avoid conflict detection function, including Self Separation Alerting, Resolution maneuvers, Omni Bands, Well Clear Recovery, and TCAS RA. VSCS will receive all DAA messages published by SaaProc as presented in sections above and will be displayed according to the requirements in the VSCS display.

8.0 Software Requirements Traceability and Implementation

The full set of software requirements for ACAS Xu FT2 is documented in the LVC SWRD-02 Rev E. All requirements in the earlier versions of the SWRD, including Rev B (IHITL), Rev C (FT3), and Rev D (FT4) have been implemented and verified. LVC SWRD-02 Rev E requirements will be implemented and subsequently verified as a part of the formal ACAS Xu FT2 V&V process.

Note that LVC SWRD-02 Rev E requirements can be traced to the system level requirements as documented for ACAS Xu FT2 in LVC SRD-01 Rev D.

9.0 Appendix A – Acronyms

| | |
|-------|--|
| ADS-B | Automatic Dependent Surveillance-Broadcast |
| AFRC | Armstrong Flight Research Center |
| AFSRB | Airworthiness and Flight Safety Review Board |
| ATC | Air Traffic Control |
| ARC | Ames Research Center |
| CCR | Configuration Change Request |
| CDR | Critical Design Review |
| CI | Configuration Item |
| CM | Configuration Management |
| CMP | Configuration Management Plan |
| COTS | Commercial Off The Shelf |
| CSD | Cockpit Situation Display |
| CSCI | Computer Software Configuration Item |
| CVS | Concurrent Versions System |
| DCP | Dryden Centerwide Process |
| DHB | Dryden Handbook |
| DOC | Document |
| DOP | Dryden Operational Procedure |
| DPR | Dryden Procedural Requirement |
| DR | Discrepancy Report |
| DSRL | Distributed Simulation Research Laboratory |
| FAA | Federal Aviation Administration |
| FMR | Flight Media Release |
| FRR | Flight Readiness Review |
| GB | Guidebook |
| GCS | Ground Control Station |
| GOTS | Government off-the-shelf |
| HLA | High Level Architecture |
| HSI | Human Systems Integration |
| ICD | Interface Control Document |

| | |
|---------|--|
| IRS | Interface Requirements Specification |
| ISRP | Integrated Systems Research Program |
| IT | Information Technology |
| IT&E | Integrated Test and Evaluation |
| IV&V | Independent Verification and Validation |
| JADEM | Java Architecture for DAA Extensibility and Modeling |
| LVC | Live Virtual Constructive Distributed Environment |
| MACS | Multi-Aircraft Control System |
| MOTS | Modifiable off-the-shelf |
| MPI | Multi-Purpose Interface |
| NASA | National Aeronautics and Space Administration |
| NPR | NASA Procedural Requirements |
| N/R | Not Required |
| OFP | Operational Flight Program |
| ORD | Objectives and Requirements Document |
| PDR | Preliminary Design Review |
| PCM | Pulse Code Modulation |
| REV | Revision |
| RFDP | Research Flight Data Processor |
| RGCS | Research Ground Control Station |
| RTCA | Radio Technical Commission for Aeronautics |
| SAP | Software Assurance Plan |
| SaaProc | Sense And Avoid Processor |
| SBU | Sensitive But Unclassified |
| SDA | Software Design Agent |
| SDD | Software Design Description |
| SDP | Software Development Plan |
| SRD | System Requirements Document |
| SWRD | Software Requirements Document |
| SRR | System Requirements Review |
| STD | Standard |
| STR | System (or Software) Test Report |

| | |
|--------|--|
| SWDF | Software Development Folder |
| TBD | To Be Determined |
| TOC | Table of Contents |
| TRR | Test Readiness Review |
| UI | User Interface |
| UML | Unified Modeling Language |
| V&V | Verification and Validation |
| V&VTP | Verification and Validation Test Plan |
| V&VTPR | Verification and Validation Test Procedure |
| VDD | Version Description Document |
| XML | Extensible Markup Language |
| | |