Treball Final de Carrera

Títol: Analysis of Interoperability between Communication UAV Networks
Titulació: Enginyeria Tècnica Aeronàutica, especialitat en Aeronavegació
Autor: Alejandro Soria Sanz
Director: Jorge Ramírez Alcántara
Data: 21 de desembre de 2010
Overview:

The main objective of this Project is to perform a compatibility study between three different communication network messages: TADIL-J's, CCSDS and AP04.

This Project firstly consists on the research of documentation concerning each communication network message. Subsequently the compiled documentation is cross-documented between communication networks for a further interoperability study. The aim is to specify similarities and dissimilarities between the messages.

Thanks to the stated processes it is then possible to implement the attestation to a java-based program, which is the responsible of showing and translating between messages. The results obtained are the fruits of the interoperability and software simulation design processes. The last part of the memory describes verification process, which makes sure that the results obtained are suitable according to the standards of the Project.

The contents of this Project are purely explanatory. It describes the established guidelines, the marked objectives, and the difficulties that appeared throughout the entire Project, the techniques to abroad the problems and the results obtained. The practical results of the work are contained in the annex of this document.
1. INTRODUCTION

1.1. Objectives of the Project ......................................................... 8
1.2. Structure of the Document ....................................................... 9

2. PROJECT CONTEXT ................................................................. 10

2.1. Work Group ............................................................................. 10
2.2. Java Eclipse Platform .................................................................. 11
  2.2.1. Creation of a Java Project ................................................... 12
  2.2.2. Eclipse Tabs ........................................................................ 12
2.3. Aeronautical Communication Datalinks ...................................... 13
2.4. OSI Reference Model ............................................................... 14
2.5. Project Focus Layer .................................................................. 15
2.6. Joint Operations ...................................................................... 16
  2.6.1. Joint Operations E.g. Search and Rescue ................................. 16
2.7. Communication Networks ......................................................... 17
  2.7.1. Tactical and Digital Link J .................................................. 18
  2.7.2. CCSDS ............................................................................... 18
  2.7.3. AP04 ............................................................................... 19

3. INTEROPERABILITY BETWEEN THREE DIFFERENT COMMUNICATION NETWORKS ........................................................................................................ 20

3.1. Message Comparison ............................................................... 20

3.1.1. TADIL-J .............................................................................. 21
  3.1.1.1. Network Management .................................................... 22
  3.1.1.2. Precise Participant Location and Identification ....................... 23
  3.1.1.3. Surveillance .................................................................... 23
  3.1.1.4. Control .......................................................................... 23
  3.1.1.5. System Status .................................................................. 24

3.1.2. CCSDS .............................................................................. 24
  3.1.2.1. Status ............................................................................ 24

3.1.3. AP04 .................................................................................. 25
  3.1.3.1. System Status and Report Packet ........................................ 25
  3.1.3.2. Control .......................................................................... 26
  3.1.3.3. Precise Aircraft Situation ................................................ 26
  3.1.3.4. Payload ......................................................................... 26

3.1.4. Results ............................................................................... 26

3.2. Field Comparison .................................................................... 27
3.2.1.Status Reports Comparison................................................................. 27
3.2.2.Control Reports Comparison.............................................................. 28
3.2.3.Positioning Reports Comparison......................................................... 29

3.3. Dissimilarities Research........................................................................ 29

4. SOFTWARE SIMULATION DESIGN.......................................................... 30

4.1. Structure of the Design......................................................................... 30

4.2. Inheritance............................................................................................. 31
  4.2.1. Inheritance Explanation..................................................................... 31
  4.2.2. Software Design Inheritance............................................................. 31

4.3. Casting.................................................................................................... 33
  4.3.1. Primitive Data Types......................................................................... 34
  4.3.2. Interoperability Software Design....................................................... 35
  4.3.3. Message Casting................................................................................ 36
    4.3.3.1. Position Messages........................................................................ 37
    4.3.3.2. TADIL-J Surveillance Message....................................................... 37
    4.3.3.3. Status Messages............................................................................ 37
    4.3.4. CCSDS Status Messages................................................................. 38
    4.3.5. Air Control Messages....................................................................... 39
    4.3.6. AP04 Payload Message................................................................... 39

5. VALIDATION AND VERIFICATION....................................................... 40

5.1. Verification Requirements...................................................................... 42

5.2. Communication Network Architecture Verification............................ 43
  5.2.1. TADIL-J Architecture Verification..................................................... 43
  5.2.2. CCSDS Architecture Verification...................................................... 43
  5.2.3. AP04 Architecture Verification.......................................................... 44

5.3. Verification Implementation................................................................. 45

5.4. Generating a Communication Network Message.................................... 47

5.5. Simulation Verification Summary.......................................................... 47

6. CONCLUSIONS....................................................................................... 49

7. BIBLIOGRAPHY....................................................................................... 51

8. FIGURES................................................................................................... 52

9. ANNEX INDEX.......................................................................................... 53
INDEX OF FIGURES

Figure 1: ICARUS Research Group................................................................. 10
Figure 2: ICARUS Research Group Aims...................................................... 10
Figure 3: EDEP Traffic Viewer screenshot.................................................. 11
Figure 4: New Java Project......................................................................... 12
Figure 5: Project Explorer.......................................................................... 13
Figure 6: Console showing incorrect execution.......................................... 13
Figure 7: Console showing correct execution............................................. 13
Figure 8: OSI Reference Model.................................................................. 14
Figure 9: Communication Networks Application Layer................................ 16
Figure 10: Search and Rescue Operations................................................... 17
Figure 11: TADIL-Js Message Catalog........................................................ 18
Figure 12: AP04 Message Catalog............................................................... 19
Figure 13: Comparison between Datalinks................................................ 20
Figure 14: TADIL-J Joint Operation............................................................. 22
Figure 15: Results Communication Network Interoperability..................... 27
Figure 16: Status Reports Comparison....................................................... 28
Figure 17: Control Reports Comparison..................................................... 28
Figure 18: Positioning Reports Comparison................................................ 29
Figure 19: Software Structure Design........................................................ 30
Figure 20: Cars Inheritance Example.......................................................... 31
Figure 21: Representation example............................................................. 32
Figure 22: Inheritance Software Design....................................................... 32
Figure 23: Primitive Data Type Casting Example (1).................................. 35
Figure 24: Primitive Data Type Casting Example (2).................................. 35
Figure 25: Representation example............................................................. 36
Figure 26: Software Design with Casting Methods...................................... 36
Figure 27: Position Report Messages Casting............................................. 37
Figure 28: Status Report Messages Casting................................................ 38
Figure 29: Control Report Messages Casting.............................................. 39
Figure 30: Aerospace Recommended Practice 4754 Verification Structure...... 40
Figure 31: V Cycle..................................................................................... 42
Figure 32: TADIL-J Architecture Verification............................................. 43
Figure 33: CCSDS Architecture Verification.............................................. 44
Figure 34: AP04 Architecture Verification.................................................. 44
Figure 35: random Message Generator.................................................... 45
Figure 36: random Generator Message Example Result............................ 46
Figure 37: random Generator Message Example Cast............................... 46
Figure 38: Gantt Initial Diagram.................................................................. 49
Figure 39: Final Gantt Diagram.................................................................. 49

INDEX OF TABLES

Table 1: Verification Matrix example......................................................... 42
Table 2: FA Verification Matrix................................................................. 48
Table 3: Software Verification Procedures FA.......................................... 48
## GLOSSARY

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP04</td>
<td>Autopilot 04</td>
</tr>
<tr>
<td>ATC</td>
<td>Air Traffic Control</td>
</tr>
<tr>
<td>CCSDS</td>
<td>The Consultative Committee for Space Data Systems</td>
</tr>
<tr>
<td>COSPAS-SARSAT</td>
<td>Cosmicheskaya Sistyema Poiska Avariynich Sudov - Search and Rescue Satellite Aided Tracking</td>
</tr>
<tr>
<td>EDEP</td>
<td>Early Demonstration &amp; Evaluation Platform</td>
</tr>
<tr>
<td>EETAC</td>
<td>Escola d'Enginyeria de Telecomunicació i Aerospatial de Castelldefels</td>
</tr>
<tr>
<td>FA</td>
<td>Failure of Analysis</td>
</tr>
<tr>
<td>HF</td>
<td>High Frequency</td>
</tr>
<tr>
<td>ICARUS</td>
<td>Intelligent Communications and Avionics for Robust Unmanned Aerial Systems</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization</td>
</tr>
<tr>
<td>QA</td>
<td>Quality Assurance</td>
</tr>
<tr>
<td>OSI</td>
<td>Open System Interconnection</td>
</tr>
<tr>
<td>PPLI</td>
<td>Precise Participant Location Identification</td>
</tr>
<tr>
<td>RELNAV</td>
<td>Relative Navigation</td>
</tr>
<tr>
<td>SAR</td>
<td>Search and Rescue</td>
</tr>
<tr>
<td>TADIL-J</td>
<td>Tactical Digital Information Link - J</td>
</tr>
<tr>
<td>TDMA</td>
<td>Time Division Multiple Access</td>
</tr>
<tr>
<td>UAS</td>
<td>Unmanned Aircraft System</td>
</tr>
<tr>
<td>UAV</td>
<td>Unmanned Aircraft Vehicle</td>
</tr>
<tr>
<td>UPC</td>
<td>Universidad Politécnica de Barcelona</td>
</tr>
<tr>
<td>VHF</td>
<td>Very High Frequency</td>
</tr>
<tr>
<td>WGS-84</td>
<td>World Geodetic System - 84</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

The actual aim of aeronautical communications is to tend to reliable and efficient communications. Actually aeronautical communications are composed of analogical voice communications; therefore the interest of using more secure communications relies on the use of digital ones.

Communications take a much more important role in UAS operations than for manned aircraft because control is done at the ground. UAS operations must be concerned on the fact that all decisions processes take at the ground either before or during the operation of the flight. This Project wants to satisfy the needs of interoperating between different communication UAS networks as an only network.

1.1. Objectives of the Project

The objectives that we pretend to perform in this Project are:

• Explain what a datalink protocol is, and what features does it provide to communication networks.
• Explain what Joint Operations are and what are their advantages.
• Understand that every communication network normally uses its type of datalink.
• Necessity to interoperate different datalinks.
• Analysis of interoperation between datalinks.
• Get to know the principle of functioning that has Java Eclipse and document on its utilization.
• Creation of a java based software.
• Process of creation and execution of a java Eclipse based Project.
1.2. Structure of the Document

The objectives of the document stated before have been structured in different chapters as shown next. This structure follows the same structure as the annexes.

In Chapter 2, we give a brief introduction about the context of the Project. We will explain the needs of doing this Project and for what research group was it done. We will also explain the purpose and key elements of aeronautical communication datalinks so that the reader is introduced into communication networks. This introduction will then help us understand better the work done throughout the Project.

In Chapter 3, interoperability process takes place. We will interoperate three different communication networks; TADIL-J, CCSDS and AP04. This will lead to the research of similarities and dissimilarities between the communication networks for a further implementation in a software design.

Chapter 4 is dedicated to the design and implementation of software that takes into notice the interoperability process between the three different communication networks. It also guides into a translation process between communication networks via castings.

In Chapter 5 we validate and verificate the Project starting with the communication networks architecture and ending with the software simulation. We will explain the simulation expectations and achievements of the Project and the different verification methods used to accquire them.

Chapter 6 brings the Projects conclusions, the initial Projects planning with its final results.
2. PROJECT CONTEXT

2.1. Work Group

Researchers of the Technical University of Catalonia form Icarus Research Group. The group is formed by permanent faculty, some researchers, PhD and Ms. Students of the Technical School of Castelldefels (Escola d’Enginyeria de Telecomunicació i Aerospacial de Castelldefels. EETAC).

“The research of the ICARUS group aims at two main objectives:

- The automation and development of on-board avionics and ground systems in order to support either manned and Unmanned Aerial Systems (UAS) while providing high levels of flexibility and low development costs.

- The improvement of Air Transportation efficiency while reducing its environmental impact and therefore, considering Air Transportation sustainability as a broad concept. In this area, we specially focus in Air Traffic Management (ATM) and aircraft operations.

In order to address our two main objectives we use extensively Information and Communication Technologies (ICT) along with methods and techniques coming from Computer Science (CS) and Operational Research (OR) disciplines. Moreover, we also focus and develop specific applications for both UAS and Air Transportation research lines”. [1]
The Projects aim inside ICARUS Research Group [1] is to interoperate three different communication networks to obtain an analysis of Interoperability. It has been done with views to make relative navigation and communication simulations. Relative Navigation (RELNAV) is used to know distance between two or more communication terminals by a measure of the arrival times of transmissions and the correlation with reported positions. The results obtained from the creation of a java based Project will be useful for a future integration in an early demonstration and evaluation platform called eDEP that ICARUS Research Group uses for its improvement research in air transport efficiency. These simulations can be then used by end users, among them, my tutor Jorge Ramirez.

2.2. Java Eclipse Platform

The aim of eDEP is to produce a low-cost, lightweight, web-enabled ATM simulator platform, offering an ideal environment for research and advanced concept projects to rapid prototype applications [2]. It has to be clarified that integration of the software in eDEP is not an objective of the Project. We also use it because its equipped with features that help tasks by making them easier such as, execution and creation of java Projects.

![EDEP Traffic Viewer screenshot](image)

During the Project we are going to use the latest version of Java Eclipse Platform, which is: Java Eclipse Helios. We use a Java based platform as Eclipse for the development of the software from the necessity for a future integration in eDEP which also is a Java based platform. Next is described how to create a Java Project and what features does it have.
2.2.1. Creation of a Java Project

“A Java Project contains source code and related files for building a Java program. It has an associated Java builder that can incrementally compile Java source files as they are changed”. [3]

To create a Java Project once Eclipse has been installed successfully you have to go to File ‣ New ‣ Java Project. Figure 4: New Java Project Shows the window that will appear.

![New Java Project](image)

2.2.2. Eclipse Tabs

This Java Project uses a large number of files; this is why it is useful to save all of them at the same location with the java files so that it is easier to modify every file. If we change one of our files, it will be enough with compiling the Project so that the file change is cropped up and this will save us a large amount of time.

For the perfect functioning of the program we have to specify the functions of it tab:

- Project Explorer. At Project Explorer you can visualize all the existing .java files which form the Project. This .java files include the code that is the spine of our interoperability Project.
• Console: At Figure 7 we can visualize all the results after running the Project simulation. It is very important because if the code generation has some sort of error the program shows error messages showing the position of the error inside the code as shown in Figure 6.

2.3. Aeronautical Communication Datalinks

In aeronautical communications, a datalink system is used to send and receive digital information between a number of aircrafts and a number of locations. Aeronautical communications are carried out using analogical voice communication. The main disadvantages of using voice communications are security breaches and that they are only communications point to point.
Nowadays communication datalinks are being implemented in aeronautical communications, because they arise many benefits for communication, as:

- Far secure than voice communications
- Ensure only authorized users view data
- Access to real-time data
- Data is structured so we can handle automatically
- Crypto-Messages

2.4. OSI Reference Model

The problem that appears in communication protocols when wanting to interlink between different protocols is that datalinks are not compatible from one to another because communication is structured in a different way because their structures are not identical. This is why in 1978; OSI reference model was proposed by H. Zimermann [4] to have a reference platform. This necessity forced the International Organization for Standards with a universally accepted platform.

The OSI Reference Model as its name indicates it's a reference model, which means that this model is not obliged to obey. It was created as a reference for the architecture of datalinks.

The next Figure shows the architecture of the OSI Reference Model:

![Figure 8: OSI Reference Model](image)

The architecture of the OSI Reference Model is structured in seven different layers as defined by H. Zimermann [4].
The Physical Layer is the first layer of the OSI Model. It is based on physical and electrical specifications. It defines the bond between a device and a transmission medium.

The Datalink Layer is the second layer of the OSI Reference Model. It is based on the interaction of multiple devices with a shared medium. It is the Layer in charge of correcting errors that may have occurred at the Physical Layer and controlling access to media.

The Network Layer is the third layer of the OSI Reference Model. The Network Layer is the layer in charge of transmission of sequences of data from a destination or multiple destinations to one or more networks, while controlling its service quality demanded by the Transport Layer.

The Transport Layer is the fourth layer of the OSI Reference Model. This layer is the one in charge of:

- Data stream support,
- Reliability,
- Flow control,
- Multiplexing.

The Session Layer is the fifth layer of the OSI Reference Model. This layer is the layer in charge of the management of sessions between users in application processes. In other words, between the transmitter and the receiver.

The Presentation Layer is the sixth layer of the OSI Reference Model. It is responsible for the displaying and formatting of the information delivered to the receiver. As an example C/C++ or Java based programming.

The application layer is the uppermost layer of the Reference Model. It provides allowable conversion between associated application processes. The slang way of referring to this layer is: application at pilots level.

### 2.5. Project Focus Layer

The Projects interoperability chapter 3. is restrained to the Application Layer. We will focus on the Application Layer of three different communication networks: TADIL-J, CCSDS and AP04. TADIL-J is a Tactical Digital Information Link, which uses Link-16, which provides improvements to existing tactical data links. CCSDS is a committee where its datalinks are used as reference and the
third datalink; AP04 is a fully integrated autopilot datalink. You can see a Figure showing the layer used of the three datalinks at Figure 9: Communication Networks Application Layer.

![Figure 9: Communication Networks Application Layer](image)

### 2.6. Joint Operations

Different organisms can only cooperate in joint operations if everything is in place to ensure flowing cooperation. Cooperation can be between different organisms (e.g. Firefighters, COSPAS-SARSAT, Civil Protection, ATC, Emergency Military Unit) with possible heterogeneity of their communication media. The ability of different organizations to conduct joint operations designates Interoperability.

Interoperability allows different organisms to work together with the only requirement of cooperating on sharing commonly procedures, regulations and infrastructures. The main point of Joint Operations is that they will be able to communicate between them. The key point about Joint Operations is that it is not needed to use same type of equipment and share common communication to acute communication with other systems.

### 2.6.1. Joint Operations E.g. Search and Rescue

Search and Rescue is an actual civil and military joint operation. In these operations multiple entities work together for a common good, a rescue. Search and Rescue is “the search for and provision of aid to people who are in distress or imminent danger”. [5]

The next Figure shows how Search and Rescue Operations work:
The functioning of Search and Rescue operations is very complex because each organism may have different types of communication. After the activation of an emergency radio beacon it sends a digital signal to Cospas-Sarsat satellites. These satellites decode the message and then they send it to a Local User Terminal, which reports to a Mission Control Center. When the Rescue coordination center receives the report it takes an action to resolve the situation, such as:

- The task of SAR units,
- Acting as SAR mission coordinator,
- Extensive monitoring,
- Investigations to determine if a search and rescue incident is occurring. [5]

This is where Joint Operations take place. Multiple entities such as air and naval forces, ATC and coastguards work together to aid the people who are in distress. Without the advantages of joint operations these types of operations would be recklessness.

2.7. Communication Networks

We are going to be concerned with three different communication networks that are going to be studied throughout the Project with the purpose of eventually creating a software simulation design. We will create this software design thanks to cross-documentation between the derived messages from the different communication networks. In the following subchapters the Projects communication networks are going to be described.
2.7.1. Tactical And Digital Link J

Tadil-J is a Tactical Digital Information Link. “Link-16 provides certain technical and operational improvements to existing tactical data link capabilities and provides some data exchange elements which the other data links lack.” [6] This datalink is actually used by the E.E.U.U. regular, air force and navy army. TADIL-J is a military based datalink which means that data was confidential and we didn’t had access to it. Therefore TADIL-J non-confidential reference guide helped us as basis.

TADIL-Js non-confidential reference guide includes its Message Catalog and it is shown in Figure 11. The communication catalogs of all three communication networks are going to be the Rosetta stone of our Project because they include the report messages that are transmitted by each communication network.

![TADIL-Js Message Catalog](image)

**Figure 11: TADIL-Js Message Catalog**

2.7.2. CCSDS

The Consultative Committee for Space Data Systems (CCSDS) [7] is a committee where major space agencies, industrial associates and observer agencies are involved as members, which provide with generic datalinks unlike the other two communication networks. CCSDS as a difference between the other two datalinks uses space communication and follows the OSI Reference Model. Due to the lack of information that brings its reference guide it was impossible to get us with its catalog message. This setback deprived us of knowing all CCSDS report messages transmitted.
2.7.3. AP04

AP04 is a datalink used by UAV Navigation’s company. “UAV Navigation is a privately owned company with one goal in mind: innovation. The company’s core business is the design and manufacture of advanced Flight Control Systems (FCS) and related hardware and software for UAV and Experimental Aviation”. [8] It is a fully integrated autopilot datalink. It means that this datalink is used to manage automatically control of an aircraft from a ground station without a pilots intervention.

<table>
<thead>
<tr>
<th>Low Level Telemetric Control</th>
<th>A2.6</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0.0</td>
<td>Initial Entry</td>
<td>A2.7</td>
</tr>
<tr>
<td>A0.1</td>
<td>Network Time Update</td>
<td></td>
</tr>
<tr>
<td>A0.2</td>
<td>System status report packet</td>
<td>A3.0</td>
</tr>
<tr>
<td>A0.3</td>
<td>Autopilot Communications</td>
<td>A3.1</td>
</tr>
<tr>
<td>A0.4</td>
<td>Connectivity Interrogation</td>
<td>A3.2</td>
</tr>
<tr>
<td>A0.5</td>
<td>Connectivity Status</td>
<td>A3.3</td>
</tr>
<tr>
<td>A0.6</td>
<td>Route Establishment</td>
<td>A3.4</td>
</tr>
<tr>
<td>A0.7</td>
<td>Debugging Analysis</td>
<td>A3.5</td>
</tr>
<tr>
<td>A0.8</td>
<td>Ground level Settings</td>
<td>A3.6</td>
</tr>
<tr>
<td>A0.9</td>
<td>Pressure Settings</td>
<td></td>
</tr>
<tr>
<td>A0.10</td>
<td>Time slot assignment</td>
<td>A4.0</td>
</tr>
<tr>
<td>Servos Control</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1.0</td>
<td>Calibration</td>
<td>A4.1</td>
</tr>
<tr>
<td>A1.1</td>
<td>Get servo PWM signal generator calibration</td>
<td>A4.2</td>
</tr>
<tr>
<td>High Level Telemetric Control</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A2.0</td>
<td>A-Family 20Hz telemetry Packet</td>
<td>A4.3</td>
</tr>
<tr>
<td>A2.1</td>
<td>A-Family 5Hz telemetry Packet</td>
<td>A4.4</td>
</tr>
<tr>
<td>A2.2</td>
<td>B-Family 5Hz telemetry Packet</td>
<td></td>
</tr>
<tr>
<td>A2.3</td>
<td>C-Family 5Hz telemetry Packet</td>
<td></td>
</tr>
<tr>
<td>A2.4</td>
<td>D-Family 5Hz telemetry Packet</td>
<td></td>
</tr>
<tr>
<td>A2.5</td>
<td>A-Family 1Hz telemetry Packet</td>
<td></td>
</tr>
<tr>
<td>GCS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A5.0</td>
<td>Antenna Tracking</td>
<td></td>
</tr>
<tr>
<td>A5.1</td>
<td>Direction/Heading</td>
<td></td>
</tr>
<tr>
<td>A5.2</td>
<td>Reports</td>
<td></td>
</tr>
<tr>
<td>A5.3</td>
<td>Antenna Calibration</td>
<td></td>
</tr>
<tr>
<td>A5.4</td>
<td>Get GCS/2 Status</td>
<td></td>
</tr>
</tbody>
</table>

Figure 12: AP04 Message Catalog
3. INTEROPERABILITY BETWEEN THREE DIFFERENT COMMUNICATION NETWORKS

Nowadays, unmanned aircraft systems are being commonly used to the foreground for civil and military organizations. Civil services, border protection agencies, and First Responders use unmanned system technology. The true contest of our interoperability comes in knowing how to aim for and use technology; how to build and operate an integrated system that gets the final user the data they need in real time. This is the main reason why we need to interoperate between these types of communication networks. The Projects interoperability as stated in the Project Focus Layer is restrained to the Application Layer.

3.1. Message Comparison

Now that we have the message catalog of all communication networks except for CCSDS we can start to find similarities. As shown in Figure 13, we are going to compare all three communication networks as to semantic and field levels. We will need to first find what type of report messages of each message catalog are suitable, and which ones are not suitable for this Project.

All three communication networks differ. TADIL-J message catalog includes both military and civil report messages. In the other hand we have AP04 and CCSDS message catalog that only includes civil report messages. The only bad news is that CCSDS catalog message does not suit our project as the other communication networks because it is based for space datalinks. For a more detailed study about the three different communication networks please read the “interoperability” annex.

![Figure 13: Comparison between Datalinks](image-url)
3.1.1. TADIL-J

TADILs Link-16 message catalog is based in military and civil transmissions as shown in Figure 11. We will summon only civil transmissions because they accommodate to the ranges of our Project. The report messages are stated below:

- Network Management,
- Precise Participant Location and Identification,
- Surveillance,
- Control,
- System Status.

The rest of TADIL-J message catalog report messages are stated below:

- Antisubmarine Warfare,
- Intelligence,
- Information Management,
- Weapons Coordination and Management,
- Platform,
- Electronic,
- Threat Warning,
- National Use.

These TADIL-J report messages are oriented to military transmissions so we will not focus on them. It is necessary to clear that although these report messages are not used throughout the Project they are used in military joint operations and they don't differ to civil joint operations. As shown in Figure 14, we can see the different types of platforms that use TADIL-J datalinks.
The next subchapters contain TADIL-J civil report messages. We had to make some assumptions about what type of data does each report message transmit because TADIL-J is a military communication network and we did not had the permission to handle confidential data. Now you will ask yourself, what is the purpose of making hypothesis of the transmitted report messages data? Well, this was done because these hypothesis where needed to support my Projects tutor thesis.

3.1.1.1. Network Management

Network Management entry is entirely about TADIL-Js Physical Layer. We will only describe it to give the reader a better idea about this communication network. It includes

- TDMA. It is the channel access method that is used in this medium network.
- Synchronization. It is the process, which requires event coordination to operate the communication network in unison.
- Relative Navigation. It is used to determine the distance between communication terminals by measuring the arrival times of transmissions and correlating them with reported positions.
- Geodetic Positioning. WGS-84.
3.1.1.2. Precise Participant Location and Identification

Each TADIL-J system periodically (Rate of actualization of each PPLI: 6 sec) transmits a Precise Position Location Information (PPLI) report. It includes:

- Secure location identification of a participant of the operation. It shows position in latitude, longitude and altitude \((x,y,z)\) with WGS-84 geodetic positioning.
- 3D positioning thanks to an aerial radar trace.

3.1.1.3. Surveillance

Each TADIL-J system periodically (Rate of actualization of each Surveillance: 12 sec) transmits a Surveillance report. It includes:

- It shows if surveillance is connected.
- Location identification of a detected trace by aerial surveillance radar. It shows position in latitude, longitude and altitude \((x,y,z)\) with WGS-84 geodetic positioning.
- Update request. It shows a message that indicates if the trace detected is from a joint operation organism.

3.1.1.4. Control

Control reports do not have an actualization rate; they are only generated case by case. However, for a Flight Plan with multiple waypoints it will be necessary to fragment the message in several transmissions (e.g. 4 transmissions spaced 6 seconds until construct the complete message).

- Waypoints. Instructions from ATC to an aircraft indicating the new flight path with multiple waypoints (the order of 12).
- Vector. Instructions from ATC to an aircraft to change flight dynamics, speed or change flight altitude.
3.1.1.5. System Status

System report statuses also do not have an actualization rate. Status includes:

- Fuel Report
- Estimated Time of Arrival,
- Estimated Time of Departure,
- Report Message,
- Fuel Report,
- Platform,
- Maximum Endurance,
- Command,
- Maximum Range,
- Payload,
- Optimum Cruise Speed,
- Equipment,

3.1.2. CCSDS

Due to the lack of information that we could get from CCSDS reference guides. The only useful report message that we could extract and that are interesting for our Project is status.

3.1.2.1. Status

CCSDS includes system report status; they also do not have an actualization rate. Status includes:

- Spacecraft Identifier
- Message Packet Ordering

The difference between the other two datalinks and CCSDS datalinks is that CCSDS is a generic datalink platform. This means that CCSDS is a provider of datalinks. By being a provider its datalinks are used as an information container that's why we don't have a catalog message and we only obtain a spacecraft identifier and a message packet ordering from a documentation process.
3.1.3. AP04

From the AP04 message catalog in Figure 12 we have extracted the next report messages:

- Low Level Telemetric Control. Where we can find:
  - System status and report packet.
- High Level Telemetric Control. Where we can find:
  - Control,
  - Precision Aircraft Situation.
- Payload.

The next subchapters contain AP04 report messages transmitted data.

3.1.3.1. System Status and Report Packet

AP04 System Status and Report Packet as TADIL-J System Status do not have an actualization rate. It is actualized when demanded by the ground station. It includes:

- Report Message,
- Command,
- Equipment,
- Estimated Time of Arrival,
- Estimates Time of Departure,
- Maximum Endurance,
- Maximum Range,
- Optimum Cruise Speed.

All of them characterize as information provided by the aircrafts autopilot to the ground station.
3.1.3.2. Control

AP04 Control reports do not have an actualization rate such as TADIL-J Control; they are only generated case by case. It also transmits the same information:

- Waypoints,
- Flight dynamics,
- Vectors.

3.1.3.3. Precise Aircraft Situation

AP04 message catalog includes precise aircraft situation that means that periodically a crypto-secure location identification of the aircraft is transmitted to the ground station. This report includes:

- Latitude,
- Longitude,
- Altitude.

All three are represented as (x,y,z) in WGS-84 geodetic positioning.

3.1.3.4. Payload

AP04 message catalog includes a payload report packet. This report includes:

- Payload.

It includes the UAV payload at the instance of the actualization. The difference between AP04 and TADIL-J regarding to payload is that AP04 has its own payload report packet in its message catalog and TADIL-J includes payload inside its status report packet.

3.1.4. Results

The results obtained after the Interoperability process between the three different communication networks are shown in Figure 15. We obtain four
messages for TADIL-J, one for CCSDS and four for AP04. These results are deeply explained at the “interoperability” annex.

![Figure 15: Results Communication Network Interoperability](image)

### 3.2. Field Comparison

These results obtained are going to be compared between them to see if the report messages transmitted are similar or dissimilar. It will then help for the creation of the java software.

#### 3.2.1. Status Reports Comparison

We know from the previous interoperability process that all three datalinks have status reports. Figure 16, shows graphically the obtained results.
As we can see both TADIL-J and AP04 status reports transmit nearly the same data. The difference between both is that AP04 does not transmit at status neither fuel, platform and payload.

### 3.2.2. Control Reports Comparison

The comparison of the control reports of TADIL-J, CCSDS and AP04 is shown in Figure 17. We can notice that CCSDS Data Links do not provide a control Report. They do not send control reports because CCSDS datalinks is used for satellite communications.

TADIL-J and AP04 datalinks transmit just the same information therefore we can state that their control reports for our Project are the same.
3.2.3. Positioning Reports Comparison

Position reports interoperability between TADIL-J and AP04 datalinks show that they transmit the same data. It has to be clarified that they transmit the same data but its use is not the same. For example TADIL-J positioning report transmits the location of the joint link 16 participants, where AP04 shows only the aircrafts positioning. What we can say is that all of them use WGS-84 so latitude, longitude and altitude is expressed as x,y and z.

![Figure 18: Positioning Reports Comparison](image)

3.3. Dissimilarities Research

We know that these three communication networks are unlike. TADIL-J is a military based Tactical Digital Information Link, which uses link-16 to provide improvements to existing tactical data links. Then we have the Consultative Committee for Space Data Systems (CCSDS), which provides with generic datalinks and follows the OSI Reference Model and finally we have AP04’s datalink used by UAV Navigation's company.

From the process of interoperability we have deduced that TADIL-J is the only datalink that contains in its catalog message a report message for Surveillance. It reports the same as PPLI report messages but instead of reporting a secure location identification of a participant of the operation it reports a secure location of a non participant.

As TADIL-J with Surveillance, AP04 transmits a report message that includes Payload. AP04 is not the only communication network that uses payload but it is the only one that transmits Payload as a report message. Instead TADIL-J sends its payload through its System Status.
4. SOFTWARE SIMULATION DESIGN

This chapter includes the phases for the creation of a software simulation implemented on a Java based program. It will explain the structure of the design, inheritance, assignment code, generation of messages and translation.

4.1. Structure of the Design

The structure of the design is composed of a Message with the three different communication networks as branches of it as shown in Figure 19. Every communication networks includes its classes. A class in object-oriented programming is a construct used for the creation of objects of the class. Each Class includes the data previously stated at the Interoperability chapter. For example if we choose TADILJ PPLI class we will know that it will include: longitude, latitude and altitude. It is shown at Precise Participant Location and Identification.

This structure will be needed when random communication messages are created.

Figure 19: Software Structure Design
4.2. Inheritance

4.2.1. Inheritance Explanation

One of the main problems faced during the generation of the software was that we couldn't manage to use a class code for another different class. The problem is that messages need to use very different types if the aim is to do an only class, although if they are done separate, they will also be incompatible. This problems was solved using inheritance.

Object-oriented programming thanks to inheritance helps classes to use code from other classes. As a simple example to understand this principle if we talk about cars, the inheritance of a class of Cars could be brands of cars such as Ford, Volkswagen and Audi, which would be the subclass of the class Cars. Therefore if these brands had also its inheritance an example of a Volkswagen inheritance could be Golf or Passat, which would also be the subclass of Volkswagen. You can see this example in Figure 20.

4.2.2. Software Design Inheritance

If we focus on our software design inheritance it will be as in Figure 22. We have the main class, which is “Message”. Then it inherits to TADIL-J, CCSDS and AP04, which also inherit to their report messages. Figure 21 represents the symbology used to illustrate classes in Figure 22.
Thanks to the inheritance representation of images explained in the above Figure 21, we obtain Figure 22.

As you can see in Figure 22, Message, TADIL-J, CCSDS and AP04 have a set interface (Type and Subtype). A set interface is where the inherited methods from the collection are contained. Message uses type because it is the message protocol, while the different communication networks use subtype because they are types of messages of the protocol. The next part of the TADIL-J class codes shows an example of the set interface.

```java
TADILJ()
{
    type = "TADILJ";
}
```
As the above code shows, TADIL-J is a type of the Message class. Therefore, CCSDS and AP04 class is also a type of the Message class.

Logically if we want to know the set interface of a TADIL-J inherited class it will be pretty similar as the code stated before. The next code shows the PPLI class, TADIL-J inherited class.

```java
PPLI()
{
    subtype = "PPLI";
}
```

The above code shows, PPLI is a subtype of the TADIL-J class.

To create a subclass inheritance, the class declaration will need to have an extension, extends keyword, followed by the name of the class to inherit from:

```java
class TADILJ extends Message {
    // Definition of TADILJ will come here
}
```

This makes TADILJ inherit the Message methods.

The advantage of using subclasses is that they inherit all the public and private members of the inherited class. These classes may contain inherited fields, which can be used directly, just like any other field. It is also possible to declare new fields in the subclass which do not belong to the class and write or declare new static or instance method in the subclasses.

4.3. Casting

Inheritance has helped us notice that each branch of the software structure design corresponds to different types of objects. The goal of this Project is to make an object of one type to be assimilable to another object of another type. Casting is the process of converting an object of one type to another.

Here is shown an example to know how to cast a type AA data into type BB data.

```java
AA a = aData;
BB b = (BB)a; // cast type AA to type BB
```

At the bottom part of the branch hierarchy of Figure 22 we have the Primitive Data Types of each object.
4.3.1. Primitive Data Types

Java language contains nine types of primitive data, which are predefined, and its name is reserved inside this language program. A primitive data type determines what variable values will contain. These are the different types:

- Byte,
- Short,
- Int,
- Long,
- String,
- Float,
- Double,
- Boolean,
- Char.

As an example of primitive data type casting we are going to explain a basic casting. Casting between an integer and a floating point.

```java
int i=0;
float f;
f=(float)i; // Cast int as float
```

The above code summarizes how an integer casts a float primitive data type. It assigns the integer number as a float.

Each different type of primitive data has its own characteristics. For example in our software design at precise position location identification for TADIL-J we use:

```java
int lat;
int lon;
int alt;

lat = 0;
lon = 0;
alt = 0;
```

We use integer primitive data as shown with latitude, longitude and altitude and then we initialize the PPLI parameters.

When having to generate equivalence between two different messages of different communication networks as shown in Figure 23 is where casting comes into play.
We have a Precise Aircraft Situation message and a precise position location identification report message. In this case, both contain the same type of data therefore casting has to be done between integers.

By having both messages with the same primitive data types casting is summarized in the exchange of the same data from one to another.

In the Project we use casting to convert data from one type of communication network to another one. It is going to be used as a translation between communication networks.

**4.3.2. Interoperability Software Design**

In chapter 4.3. we introduced castings to the software design. In this chapter we are going to describe each message casting. To introduce it, Figure 25 represents the simbology used to illustrate Figure 26:
Here we can see that PPLI contains Latitude, Longitude and Altitude as attributes. As methods it shows that the generator is using it and it casts AP04 precise aircraft situation (PAS) message.

The Following Figure represents the final software design with its castings:

![Software Design with Casting Methods](image)

**Figure 26: Software Design with Casting Methods**

### 4.3.3. Message Casting

In this chapter, description of each message casting is going to take place. As shown in Figure 26, we deduced that these datalinks had report messages in common. Now in the software simulation design thanks to casting we manage the process of converting an object of one type to an other. The use of casts
will lead to a translation process between each communication network introduced by the Interoperability Process.

4.3.3.1. Position Messages

From the Precise Participant Location and Identification and the Precise Aircraft Situation interoperability chapter we can deduce that both position messages cast each other. This structure is shown in Figure 27. As latitude, longitude and altitude of a TADIL-J and AP04 communication network is represented in WGS-84, therefore its communication network translation process will only consist in representing the position obtained in the other communication network without any modification, just the same as Figure 24.

![Figure 27: Position Report Messages Casting](image)

4.3.3.2. TADIL-J Surveillance Message

Surveillance message will not cast any other message despite it reports the same parameters as Position Messages. This is due because it shows the position of a non-joint operation organism, while PPLI and PAS report the position of a joint operation organism.

4.3.3.3. Status Messages

To implement the interoperability from the TADIL-J System Status and AP04 System Status and Report Packet chapters we propose Figure 28 structure. It shows both TADIL-J and AP04 status messages. Due to the lack of information about CCSDS we could not find any resemblance with the other two communication networks. We can see that mainly all the messages of both report messages cast each other.
We can see that the following messages cast between both communication networks.

- Fuel,
- Report Message,
- Equipment,
- ETD & ETA,
- Max. Endurance,
- Max. Range,
- and Optimum Speed.

The communication network translation process for these messages is left for a further project development by another student. It will be needed to recognize which type of messages does each message contain. For example how is the Fuel message reported in lbs., kg, %...

Figure 28: Status Report Messages Casting

4.3.4. CCSDS Status Messages

CCSDS status message will not cast any other message. This is due because CCSDS reference guide lacks of information referred to its status message.
4.3.5. Air Control Messages

From the interoperability chapter we can see that all the messages of both air control messages cast each other. In Figure 29 you can see both TADIL-J and AP04 control report messages.

As in Status Messages the communication network translation process is left for a further project development by another student. This is because the communication reference guides lack of information about the type of data inside these messages.

![Figure 29: Control Report Messages Casting](image)

4.3.6. AP04 Payload Message

AP04 payload message will not cast any other message despite it reports a payload message as TADIL-J status message.
5. VALIDATION AND VERIFICATION

This chapter includes the validation and verification process that have taken place during the Project. For a more detailed study please go to annex 4 “Verification Procedures”. We have verified both the architecture and the results of the software simulation implemented on a Java based program.

Verification as validation is a process that takes place all along the project. When verifying you have to ask yourself, are we doing the Project that we were intended to do and in a correct way? Instead when talking about validation you ask yourself, are we doing the correct product, or the one the client wants?

Verification process starts verifying the software design that had its own design requirements. After the verification process methods a verification summary is obtained. The next Figure, Figure 30 shows the process to be done to verify the Project that comes from ARP 4754.

Verification is a process that establishes that each task of the project is being done correctly. In the next subchapter Communication Network Architecture Verification, we show our Project verification process.
Verification process:

- Confirms that the intended functions have been correctly implemented.
- The requirements have been satisfied.

This process consists of four verification methods:

- Analysis,
- Test,
- Previous experiences,
- Mathematical demonstrations.

During the process of verifying intended functions, any anomalies recognized should be reported so that they can be reviewed.

In this Project, we are only going to use analysis and test verification methods because we don't have any previous experiences neither formulas throughout the Project.

If we focus on validation, in this Project the validation client has been my tutor Jorge Ramirez. Validation takes place as shown in Figure 31, at the V cycle diagram in an upper direction.

The V Cycle diagram of Figure 31 represents the Projects structure. We started with an Interoperability chapter after referencing our Projects context situation. We then have our specification similarities research that lead to a software simulation design. When the software simulation design was finished we implemented it in a Java based program called Eclipse. You can find the software code at the annex “software design code”.
5.1. Verification Requirements

At the beginning of the Project, concisely at the planning process we stated our verification requirements. The requirements were to generate a random message which would show all types of report messages that could be received by the communication networks. After the reception of the report messages we had to be able to distinguish between the three different types of communication network data; TADIL-J, CCSDS and AP04. These requirements are listed in the verification matrix at the “verification procedures” annex. Table 1 shows an example of a verification matrix. This example includes five different procedures with the verification method that is applied in each case.

<table>
<thead>
<tr>
<th>Procedures</th>
<th>Verification Method Applied</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Analysis</td>
</tr>
<tr>
<td>Does AP04 Precise Aircraft Situation</td>
<td></td>
</tr>
<tr>
<td>Includes lat, lon and alt?</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Analysis</td>
</tr>
<tr>
<td>3</td>
<td>Analysis</td>
</tr>
<tr>
<td>Does CCSDS Report Message include Id and Packet?</td>
<td></td>
</tr>
</tbody>
</table>
| 4          | Test                        | 1
| Does PPLI extends TADIL?             |                       |
| 5          | Test                        | 1
| Is latitude random aRandom.nextInt(180) + -90? |                       |

*Table 1: Verification Matrix example*
5.2. Communication Network Architecture Verification

In order to meet the standards of the Project we have inspected and reviewed the communication network catalog messages in order to establish that the Project is well based.

5.2.1. TADIL-J Architecture Verification

We have verified TADIL-J catalog Message to establish that the results obtained are the ones being studied during the Project.

TADIL-J Message Catalog and the results obtained at the Interoperability process are shown in Figure 32. Verification established that the results were the same as the ones withdrawn at the Tactical And Digital Link J Interoperability process. You can see the results at the “Verification Procedures” Annex. This means that TADIL-J has been verified successfully.

![TADIL-J Architecture Verification](image)

5.2.2. CCSDS Architecture Verification
We did also verify CCSDS architecture results and as TADIL-Js, the results obtained were successful. Therefore Figure 33 shows CCSDS results, that thanks to verification, we know that they are the same as in the Interoperability process.

5.2.3. AP04 Architecture Verification

We have also verified AP04 catalog Message to establish that the results obtained are the ones being studied during the Project.

The report messages obtained after the architecture verification process are the same as the Interoperability results, they are shown in Figure 34. We can state that all three communication networks architecture have been verified and that they agree with the Interoperability process. These results are available at the “Verification Procedures” Annex.
5.3. Verification Implementation

We managed to implement all communication networks as classes in Java and thanks to inheritance we could bond all of them. We also managed to make the message generator to become random so that the information achieved from it could be random. We were capable of knowing from which type of communication network was the data achieved from the message generator.

The next Figure, Figure 35, comes to represent a graphical explanation of how we achieve random the report messages of the communication networks. Thanks to random functions this can be obtained. The main point of the random message generator is that we dont know which report message is going to be the resultant.

Figure 35: random Message Generator
The resultant report messages obtained at the message generator are the messages that our simulation system is going to receive. The next step would be to translate them from one communication network to another one thanks to casings. Figure 36 represents an random Generator Message Example Result.

![Figure 36: random Generator Message Example Result](image)

From the above Figure example, we have received seven random report messages. We have received three TADIL-J and AP04 report messages and the only CCSDS report message. The software simulation will recognize each type and it will cast them to another communication network as explained at Message Casting. The result is shown in Figure 37.

![Figure 37: random Generator Message Example Cast](image)

AP04 Payload, and status CCSDS do not cast any other report messages.
5.4. Generating a Communication Network Message

So far we have designed and implemented in Java the software simulation design. We have also described each report message and have matched them using castings. Therefore we only needed to generate a structure which deals with the generation of types and subtypes. The process by which the message generator creates the report messages must be random.

As an example of obtaining a random report message it is going to be explained with PPLI message generation code:

```java
Random aRandom = new Random();
PPLI mppli = new PPLI(aRandom.nextInt(180) + -90,
                        aRandom.nextInt(360) + -180,
                        aRandom.nextInt(30000) + 0);
```

We use casting to asign to the variable aRandom an instance of a stream of pseudorandom numbers. We can see that from the example, the data to be generated is going to be the PPLI report.

```java
aRandom.nextInt(180) + -90
```

This code represents a random integer that is going to be assigned to PPLI's latitude parameter. Latitude goes from the South Pole, which has latitude of 90° south (90°S or -90°) to the North Pole, which has latitude of 90° north (90°N or 90°). Therefore we can see that the code is going to generate a random data from -90 with a range of 180, which means that it will go from -90° – 90°.

5.5. Simulation Verification Summary

In Verification Procedures at the Annex we can see the 37 steps that have been verified to see if the software design is correctly done. This is done with a verified matrix and then with a checklist so that it is easier for a reader to read. The first table contains the procedures to be verified, while the second table contains the items to be verified, the result obtained, the verification starting time and the person in charge of the quality assurance verification. Table 2 shows the procedures of the items that gave failure when verified.
The next table, table 3, shows the five items that had a failure as a result of the verification process:

<table>
<thead>
<tr>
<th>Item to Verify</th>
<th>Result</th>
<th>Time</th>
<th>QA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Java classes must have less than 150 words for better comprehension</td>
<td>FAILURE</td>
<td>11:00 24/11/10</td>
<td>Alejandro Soria</td>
</tr>
<tr>
<td>3 Message must include timestampemission and timestampandreception and imessage</td>
<td>FAILURE</td>
<td>12:00 24/11/10</td>
<td>Alejandro Soria</td>
</tr>
<tr>
<td>29 Generator implements imessage</td>
<td>FAILURE</td>
<td>11:00 25/11/10</td>
<td>Alejandro Soria</td>
</tr>
<tr>
<td>30 Generator message randomly</td>
<td>FAILURE</td>
<td>17:00 26/11/10</td>
<td>Alejandro Soria</td>
</tr>
<tr>
<td>37 Translations</td>
<td>FAILURE</td>
<td>15:30 27/11/10</td>
<td>Alejandro Soria</td>
</tr>
</tbody>
</table>

Table 3: Software Verification Procedures FA

As a result of Table 3, we can see at the “Verification Procedures” annex, that all of them except FA-5 where resolved.

The only item that could not be managed was the communication network translation process. It has been thought to be left for a further project development by an other student.
6. CONCLUSIONS

Planning was the first document created of the Project with the purpose of helping the reader understand better its structure. The aim of this document was to use it as a guideline of the the structure of the final project. You can find the planning of the Project at the “Planning” annex.

We used Gantt diagrams to illustrate the Projects planning. Figure 38 shows the Initial Gantt diagram that was done before starting with the Project.

![Figure 38: Gantt Initial Diagram](image)

The last thing that was done of the Project was the Final Gantt diagram. This was done to illustrate how the Project had evolved since the beginning and then to compare it with the suggested initial planification diagram.

![Figure 39: Final Gantt Diagram](image)

If you compare the above Gantt diagrams you can see that both diagrams differ. The Project was considered to be done during summer holidays but finally it
could not be done due to technical problems with the svn server. The svn server is the tool where we upload the Project information so that the tutor and the student can exchange documents. This made the Project to end later. Although the changes, the overall philosophy of the Project has been maintained.

The Projects software shows how managing communication translations between different types of datalinks instantly without having to cope to communicate separately between those communication network could save time. The Software manages to unify joint operations communication, which in my opinion will ensure a reduction in time.

The future of the Project will consist of the following of this Project by another student to achieve an integration in an early demonstration and evaluation platform to make relative navigation and communication simulations as a researcher for ICARUS Research Group.
7. BIBLIOGRAPHY


http://www.eclipse.org/


[7] public.ccsds.org/publications/archive/102x0b5s.pdf

[8] SAE ARP4754, Aerospace Recommended Practice.
8. FIGURES

[Figure 1]
http://icarus.upc.edu/

[Figure 2]
http://icarus.upc.edu/research

[Figure 3]
http://www.eurocontrol.fr/projects/edep/

[Figure 4]
Java Eclipse SDK Version: 3.6.1 Build id: M20100909-0800

[Figure 5]
Java Eclipse SDK Version: 3.6.1 Build id: M20100909-0800

[Figure 6]
Java Eclipse SDK Version: 3.6.1 Build id: M20100909-0800

[Figure 7]
Java Eclipse SDK Version: 3.6.1 Build id: M20100909-0800

[Figure 8]

[Figure 10]

[Figure 11]

[Figure 12]
AP04 Reference Guides.

[Figure 14]
9. ANNEX INDEX

Annex 1 --> “Planning”
Annex 2 --> “Interoperability”
Annex 3 --> “Software Design Code”
Annex 4 --> “Verification Procedures”