# UAV Data Exchange Test Bed for At-sea and Ashore Information Systems


Yannick Allard
Hugues Demers
Michel Mayrand
Dan Radulescu

*Prepared By:* OODA Technologies Inc.
4891 Grosvenor
Montréal (Qc), H3W 2M2
514.476.4773

*Prepared For:* Defence Research & Development Canada, Atlantic Research Centre
9 Grove Street, PO Box 1012
Dartmouth, NS
B2Y 3Z7
902-426-3100

Scientific Authority: Anthony Isenor
Contract Number: W7707-115137 and W7707-145677
Call Up Number: 11 (4501076100) and 3 (4501133302)
Project: 01jq, UAV Information Exchange and Exploitation
Report Delivery Date: December 2, 2014



Contract Report
DRDC-RDDC-2014-C297
December 2014

This page is intentionally left blank.

The Royal Canadian Navy (RCN) has articulated the need for more effective command teams through the improved use of information. In particular, the RCN indicates the need for improved information management techniques to promote greater situational awareness. In the maritime domain, improvements could be realized through more effective use of data collected by sensors and receivers onboard Unmanned Aerial Vehicles (UAV).

UAVs provide numerous capabilities such as persistent over-the-horizon surveillance, target acquisition, and reconnaissance. However, the data collected from a UAV will be best utilized when integrated with other information that taken collectively supports Maritime Domain Awareness.

This call-up explores the potential use of UAV collected data in both at-sea and ashore information systems. The basic idea is to establish a test bed specifically for testing UAV data exchange with at-sea and ashore information systems.

Data exchange will be investigated using instances of Coalition Shared Data Server (CSD). The CSD is an implementation of the STANAG 4559 to ensure data discovery and interoperability amongst mission participants. The testbed will evaluate its synchronization capabilities in a low bandwidth environment and explore the potential of using the CSD to store and retrieve AIS data.

In addition, a review and comparision of the current GCCS-M architecture and functionnalities with the upcoming MTC2 system is provided. A survey of UAV current and future sensor payloads is also included.

This page is intentionally left blank.

# Contents

**Contents** iii

**List of Figures** vii

**List of Tables** ix

**1 Introduction** 1

**2 Coalition Shared Data Server** 3

2.1 Coalition Shared Data Server Description . . . . . 3

2.2 Coalition Shared Data Server and interoperability . . . . . 4

2.2.1 NATO Standard ISR Library Interface . . . . . 4

2.2.2 NATO Secondary Image Format : STANAG 4545 . . . . . 5

2.2.3 NATO Ground Target Indicator Format Standard NATO Agreement (STANAG) 4607 . . . . . 7

2.2.4 NATO Motion Imagery Format . . . . . 7

2.2.5 NATO Tactical Data - Link16 format . . . . . 8

2.2.6 Other important NSIL data model . . . . . 9

2.2.6.1 NSIL Message View . . . . . 9

2.2.6.2 NSIL Report View . . . . . 9

2.2.6.3 NSIL Collection Coordination and Information Requirements Management . . . . . 10

2.3 Coalition Shared Data Server, NPR and SC2PS Installation . . . . . 10

2.4 UAV Data Upload . . . . . 12
2.5 Data Exchange between Coalition Shared Data Server instances . . . . . 12
2.5.1 Coalition Shared Data Servers Synchronization . . . . . 12
2.5.1.1 Synchronisation setup . . . . . 13
2.5.2 Slowdown Mechanism . . . . . 15
2.5.2.1 Kernel space . . . . . 15
2.5.2.2 User space . . . . . 16
2.5.2.3 Advantages and disadvantages of both approaches . . . . . 16
2.5.2.4 The router . . . . . 17
2.5.2.5 NetLimiter . . . . . 20
2.6 Data Exchange Behavior in a network of Coalition Shared Data Server instances . 21
2.6.1 Scenario 1 : Connection failure before resynchronization . . . . . 22
2.6.2 Scenario 2 : Connection failure during synchronization . . . . . 24

**3 CSD Data Visualization 27**

3.1 Visualization using SC2PS . . . . . 27
3.1.1 Connect to a CSD with SC2PS . . . . . 28
3.1.2 Upload Motion Imagery from UAV with SC2PS . . . . . 28
3.1.3 Query the CSD and visualize Motion Imagery with SC2PS . . . . . 28
3.1.4 Exporting video / imagery exploitation product to the CSD . . . . . 29
3.2 Visualization using NASA World Wind . . . . . 30
3.3 Visualization using Quantum GIS . . . . . 32
3.3.1 The CSD plugin . . . . . 33
3.3.2 Pros and cons of QGIS . . . . . 33
3.3.2.1 Developing the plugin . . . . . 34
3.4 Remarks on Visualization . . . . . 34
3.4.1 Interaction with the CSD . . . . . 34
3.4.2 Extracting reports from the CSD . . . . . 35

**4 Coalition Shared Data Server and the Global Positioning Warehouse** **37**

4.1 Naval Position Repository Installation . . . . . 37

4.2 CSD and NPR data structures comparison . . . . . 37

4.3 AIS and the CSD . . . . . 38

4.3.1 AIS Decoder Implementation . . . . . 38

4.3.2 NIEM XML exchange standard . . . . . 38

4.3.3 AIS Data to CSD . . . . . 39

4.4 Data Exchange between CSD and NPR . . . . . 41

**5 Maritime Tactical Command and Control** **43**

5.1 Global Command and Control System - Maritime (GCCS-M) / DII COE . . . . . 43

5.1.1 GCCS-M / COE Concept and Architecture . . . . . 44

5.1.2 The COE concept . . . . . 44

5.1.3 The COE architecture . . . . . 45

5.1.3.1 The COE kernel . . . . . 45

5.1.3.2 The COE Infrastructure services . . . . . 46

5.1.3.3 The COE Common Support Application services . . . . . 46

5.1.4 Data access and sharing . . . . . 47

5.1.5 Functionalities . . . . . 48

5.2 Maritime Tactical Command and Control . . . . . 48

5.2.1 Architecture of the MTC2 - The *as a Service* technology . . . . . 49

5.2.2 Data as a Service - Net-Centric Enterprise Services data strategy . . . . . 50

5.2.2.1 Community of Interest . . . . . 52

5.2.3 Infrastructure and Platform as a Service : The Consolidated Afloat Networks and Enterprise Service (CANES) . . . . . 53

5.2.4 Software as a Service . . . . . 55

5.3 Comparison of MTC2 and GCCS-M . . . . . 56

**6 UAV Sensors and Recommendations** **59**

6.1 UAV Sensor Payload . . . . . 59
6.1.1 Imagery Intelligence . . . . . 60
6.1.1.1 Video/Electro-Optic/Infrared (EO/IR) Sensors . . . . . 60
6.1.1.2 Synthetic Aperture Radar (SAR) . . . . . 62
6.1.2 Signal Intelligence . . . . . 63
6.1.3 Automatic Identification System Receiver . . . . . 64
6.1.4 Maritime Patrol Radar . . . . . 65
6.1.5 Light Detection and ranging . . . . . 65
6.1.6 Laser Radar . . . . . 66
6.1.7 Sensors in the context of maritime surveillance operations . . . . . 66
6.1.8 Sensors in the context of maritime environmental operations . . . . . 67
6.1.9 Sensor and information products in relation to the CSD capabilities . . . . 69
6.1.10 Summary and Recommendations for data simulation . . . . . 69

**7 Difficulties and Conclusion** **71**
7.1 Difficulties encountered . . . . . 71
7.2 Conclusion . . . . . 71

**Bibliography** **73**

**A Summary of UAV current sensor payload in the US Navy** **A-1**
A.1 Example of currently fielded EO/IR payloads . . . . . A-1
A.2 Example of currently fielded SAR payloads . . . . . A-1
A.3 Example of currently fielded SIGINT payloads . . . . . A-2
A.4 Example of currently fielded AIS payloads . . . . . A-2
A.5 Example of currently fielded MPR payloads . . . . . A-2
A.6 Sensor payload in development . . . . . A-3
A.7 Summary of UAV ant their current payloads . . . . . A-3

# List of Figures


2.1 STANAGs based interoperabilty. . . . . 4

2.2 NATO secondary image format structure. . . . . 6

2.3 WAN interface configuration . . . . . 18

2.4 QoS configuration . . . . . 19

2.5 Limiting bandwidth . . . . . 20

2.6 NetLimiter 3 Pro . . . . . 21

2.7 Initial state of CSD network . . . . . 22

3.1 NWW interface with AIS contacts . . . . . 31

3.2 CSD connection configuration window . . . . . 32

3.3 UAV coverage and video list . . . . . 33

3.4 Video playing in Windows Media Player . . . . . 34

3.5 QGIS with multiple layers shown . . . . . 35

3.6 CSD plugin . . . . . 36

3.7 CSD reports shown . . . . . 36

4.1 AIS messages flow . . . . . 40

5.1 GCCS-M / COE architecture. . . . . 45

5.2 MTC2 as a Service architecture. . . . . 49

5.3 MTC2 as a Service architecture. . . . . 50

5.4 MTC2 Cloud-based CONOPS. . . . . 51

5.5 MTC2 Cloud-based CONOPS in case of DoS. . . . . 52

6.1 Still Imagery Technology Forecast. . . . . 61

6.2 Videos Technology Forecast. . . . . 62

6.3 Radar Imagery Technology Forecast. . . . . 63

6.4 Signal Intelligence Technology Forecast . . . . . 64

6.5 Ocean weather image presenting waves heights and wind speed and direction . . . 68

# List of Tables

5.1 Role of enterprise and COI towards data strategy goals . . . . . 53

6.1 Current and near-term capabilities of sensor types . . . . . 69

A.1 Currently deployed UAV and available sensor payload . . . . . A-3

This page is intentionally left blank.

| | |
|---|---|
| **AIS** | Automatic Identification System |
| **API** | Application Programming Interface |
| **BAMS** | Broad Area Maritime Surveillance |
| **C4ISR** | Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance |
| **CANES** | Consolidated Afloat Networks and Enterprise Service |
| **CCD** | Charge-Coupled Device |
| **CI** | Computer Infrastructure |
| **COE** | Common Operating Environment |
| **COI** | Community Of Interest |
| **CSD** | Coalition Shared Data Server |
| **DaaS** | Data as a Service |
| **DEM** | Digital Elevation Model |
| **DII** | Defence Information Infrastructure |
| **DMS** | Discovery Metadata Specifiations |
| **ELINT** | Electronic Intelligence |
| **EO** | Electro-Optical |
| **EO/IR** | Electro-Optic/Infrared |
| **ESM** | Electronic Support Measures |
| **GCCS** | Global Command and Control System |
| **GCCS-M** | Global Command and Control System - Maritime |
| **GIG** | Global Information Grid |
| **GIOP** | General Inter-ORB Protocol |
| **GMTI** | Ground Moving Target Indicator |
| **GPW** | Global Positioning Warehouse |
| **HSI** | Hyperspectral Imagery |
| **IaaS** | Infrastructure as a Service |
| **IMINT** | Imagery Intelligence |
| **IMO** | International Maritime Organization |

| | |
|---|---|
| **IR** | Infra-Red |
| **ISR** | Intelligence Surveillance and Reconnaissance |
| **ISS** | Integrated Sensor Suite |
| **LADAR** | Laser Radar |
| **LIDAR** | Light Detection And Ranging |
| **MALE** | Medium Altitude - Long Endurance |
| **MAJIIC** | Multi-intelligence All-source Joint ISR Interoperability Coalition |
| **MASINT** | Measurement and Signatures Intelligence |
| **MI** | Motion Imagery |
| **MMSI** | Maritime Mobile Service Identity |
| **MMSS** | Multimode Sensor/Seeker |
| **MPR** | Maritime Patrol Radar |
| **MP-RTIP** | Multi-Platform Radar Technology Insertion Program |
| **MSI** | Multispectral Imagery |
| **MSARI** | Maritime Situational Awareness Research Infrastructure |
| **MTC2** | Maritime Tactical Command and Control |
| **MTI** | Moving Target Indicator |
| **NCES** | Net-Centric Enterprise Services |
| **NMEA** | National Marine Electronics Association |
| **NIEM** | National Information Exchange Model |
| **NPR** | Naval Position Repository |
| **NSIF** | NATO Secondary Image Format |
| **PA** | Project Authority |
| **PaaS** | Platform as a Service |
| **PolSAR** | Polarimetric SAR |
| **QoS** | Quality of Service |
| **RCN** | Royal Canadian Navy |
| **SaaS** | Software as a Service |

| | |
|---|---|
| **SAR** | Synthetic Aperture Radar |
| **SC2PS** | Sensor Command and Control Planning Suite |
| **SHADE** | Shared Data Engineering |
| **SIGINT** | Signal Intelligence |
| **SQL** | Structured Query Language |
| **STANAG** | Standard NATO Agreement |
| **TCPED** | Tasking, Collection, Processing, Exploitation and Dissemination |
| **UA** | Unmanned Aircraft |
| **UAV** | Unmanned Aerial Vehicle |
| **UAS** | Unmanned Aerial System |
| **XCOP** | Extensible Common Operating Picture |
| **XML** | Extensible Markup Language |

This page is intentionally left blank.

# Part 1

# Introduction

The Royal Canadian Navy (RCN) has articulated the need for more effective command teams through the improved use of information. In particular, the RCN indicates the need for improved information management techniques to promote greater situational awareness. In the maritime domain, improvements could be realized through more effective use of data collected by sensors and receivers onboard Unmanned Aerial Vehicle (UAV).

UAVs provide numerous capabilities such as persistent over-the-horizon surveillance, target acquisition, and reconnaissance. However, the data collected from a UAV will be best utilized when integrated with other information that taken collectively supports Maritime Domain Awareness.

Given the increasing role UAVs are going to play as part of wide area maritime surveillance, it is important to assess the interoperability of the data they provide and their compatibility with existing or planned data infrastructure.

This call-up explores the potential use of UAV collected data in both at-sea and ashore information systems. The basic idea is to establish a test bed specifically for testing UAV data exchange with at-sea and ashore information systems.

Data exchange will be investigated using instances of Coalition Shared Data Server (CSD). The CSD is an implementation of the Standard NATO Agreement (STANAG) 4559 ([4]) that ensures data discovery and interoperability amongst mission participants. The testbed will evaluate its synchronization capabilities in a low bandwidth environment. In addition, it will explore the potential of using the CSD to store and retrieve AIS data, which may be an important part of the data collected by UAV and other surveillance assets in the maritime domain.

This test bed will also explore the feasibility of sharing UAV data between a CSD and Naval Position Repository (NPR) instances. This will provide an assessment of the compatibility of data provided by a UAV platform to existing ashore information systems. Video and AIS data will be used as a starting point for this assessment.

This document, which is the final report for Call-up 11, is organized as follow :

- Section 2 presents a description of the CSD and the standards on which it is based. It de-

scribes the CSD installation steps, its synchronization capabilities and the slowdown mechanism that was put in place to control network connection bandwidth between CSD instances. It also includes observations about the CSD behaviour when use in a network where multiple CSD instances are present.

- Section 3 presents the different visualization solutions that were investigated to support the CSD products.
- Section 4 presents a high level comparison of the structure of CSD database with the Global Positioning Warehouse (GPW) v2.0 (or NPR) database. It highlights the difficulties of storing video data feed obtained by a UAV in the NPR database, given the version that was used as part of this Call-up. It also contains the description of the work that was performed to feed the CSD with AIS message, under a National Information Exchange Model (NIEM) XML format, retrieve them using the CSD metadata query interface, and store them in a NPR instance.
- Section 5 presents a review of the Common Operating Environment (COE) and the Maritime Tactical Command and Control (MTC2) at the architectural and functional level as the CSD is foreseen to support the data sharing and access strategy of the MTC2 and as such, the future version of the Global Command and Control System - Maritime (GCCS-M).
- Section 6 presents a review of available UAV sensor payloads, the information that can be obtained from them and the relation of the sensor output to the CSD standard where applicable.
- Section 7 presents the difficulties encounter during the realisation of this Call-up and the general conclusion of this document.

# Part 2

# Coalition Shared Data Server

This section presents a description of the CSD and the standards on which it is based. It also presents the CSD installation steps, its synchronization capabilities and the slowdown mechanism that was put in place to control network connection bandwitdh between CSD instances.

## 2.1 Coalition Shared Data Server Description

The CSD is a server implementation of the STANAG 4559 NATO standard Intelligence Surveillance and Reconnaissance (ISR) library interface. The standard Application Programming Interface (API) enable access to share repository of Ground Moving Target Indicator (GMTI), Synthetic Aperture Radar (SAR), Electro-Optical (EO) and Infra-Red (IR) imagery, Motion Imagery (MI) as well as exploitation products and other relevant data that is generated and used by Multi-intelligence All-source Joint ISR Interoperability Coalition (MAJIIC) 2 collection assets and exploitation systems.

The CSD concept was developed to address issues relating to availability and operational use of MAJIIC 2 data namely :

1. Members may use the CSD to initiate their national system and complement it with information from coalition members systems;
2. The CSD can be used even if it suffered from communication failure as it will recover by synchronizing back;
3. The CSD aims at the generation of the Common Ground Picture by providing access to GMTI, SAR, EO and IR imagery, MI and historical data;
4. The CSD removes the reliance on broadcast mechanism. Such a mechanism can place a huge strain on the communication network by enabling information request and subscription to data/information of interest.

## 2.2 Coalition Shared Data Server and interoperability

The data and information, as well as their describing metadata, which are stored within the CSD all follow a given standard to ensure interoperability between coalition members. Figure 2.1 shows the different NATO standards used within a compliant CSD.

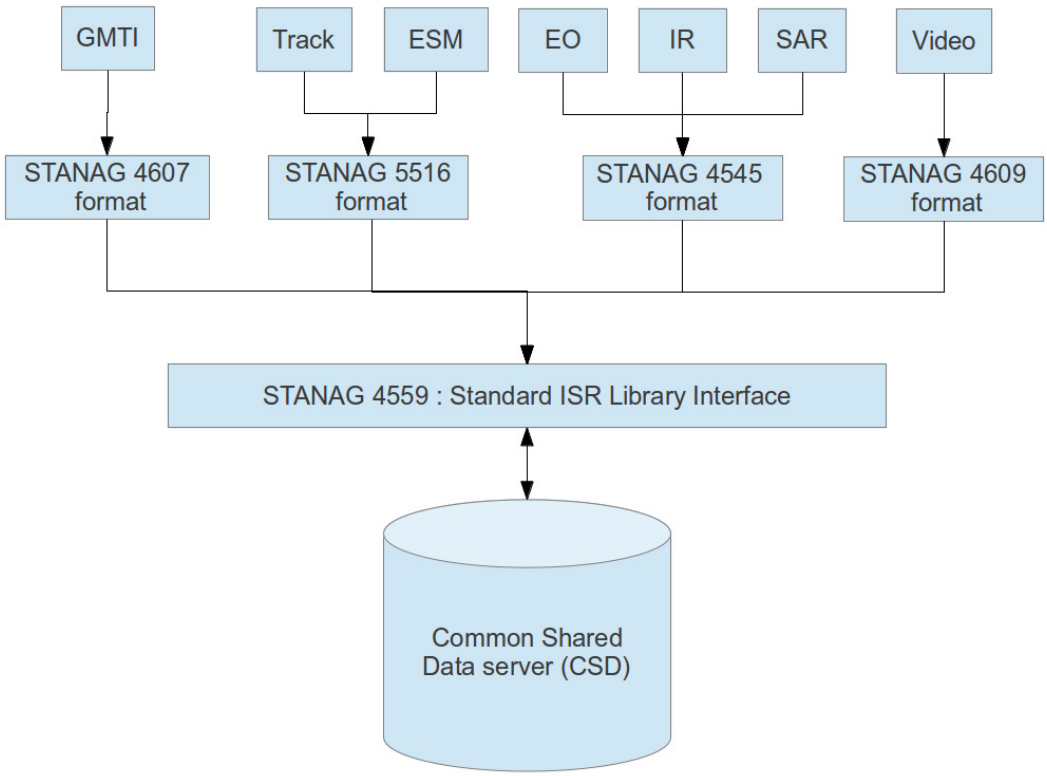


Figure 2.1: STANAGs based interoperabilty.

The following sections give the reader a brief overview of the different STANAGs involved in the current CSD implementation.

### 2.2.1 NATO Standard ISR Library Interface

The STANAG 4559, the NATO Standard ISR Library Interface (NSILI), provides a standard for accessing ISR libraries, reconnaissance databases, and product libraries of participating nations. It defines an interoperable interface to each participating Nation's ISR library system, without altering the internal architecture of any individual system. The basic concept of NSILI is for the nations to place their ISR products on their own National server, and to make those products available through the standard interface defined in STANAG 4559 to applications or users requiring that information [4].

The interface provides electronic search and retrieval capabilities for distributed users to find products from distributed libraries in support of, but not limited to, rapid mission planning and operation, strategic analysis, and intelligence preparation of the battlefield. Product Libraries and the NSIL Interface are viewed as a key standards-based technology utilized within existing Request for Information (RFI) procedures [4].

Within the CSD, formats for ISR data are defined that include NATO Secondary Image Format (NSIF) (STANAG 4545) for multiple still images, text and graphics segments, and the relative orientation of each with respect to the other segments of the image; NSIF also includes provisions for additional types and volumes of data that were not anticipated at the time the format standard was created. Because of this continued expansion of applications of NSIF, the NSILI has been seen to provide a discovery and retrieval capability for a variety of ISR data types. The flexibility of NSILI has been proven with ISR Ground Moving Target Indicator (STANAG 4607) and Digital Motion Imagery (STANAG 4609), which serve to prove the broader utility of NSILI [4].

The NSILI Client Application provides the user interface between a library and NATO users, and allows the client to use input from the user to discover a library server, search the library holdings of various servers discovered, order products, and have digital data transmitted to the user [4].

In order to query for data or information products, the user relies on metadata attached to the said product. There may be two types of metadata supported. These are [4]:

1. Mandatory metadata. All servers are required to support queries involving these metadata and should make these metadata available where possible. They are described later in this section.
2. Optional, metadata. These are metadata that are not required by the NSIL Interface. Using the discovery mechanism provided by the NSIL Interface, Clients can discover the optional metadata attributes supported by a particular library.

### 2.2.2 NATO Secondary Image Format : STANAG 4545

The NSIF (2.2) describes the format of digital images, Computer Graphic Metafile (CGM) graphics, text data and metadata that may be present within the NSIF file. It does not define the image graphic or text requirements of the host system. The host system is responsible for the handling of unpacked image, graphic, and text data, as well as image, graphic and text display capabilities [5].

Though the STANAG 4545 was conceived initially to support the transmission of a file composed of a single base image, image insets (subimage overlays), graphic overlays, and text, the format makes it suitable for a wide variety of data file exchange needs. One of the flexible features of the NSIF is that it allows several Segments to be included in one NSIF File, yet any of the data types may be omitted. Thus, for example, the NSIF may equally well be used for the storage of a single portion of text, a single image or a complex composition of several images, graphics, and text. Information on other implementations can be found on the NSIF Registry on the Internet [5].

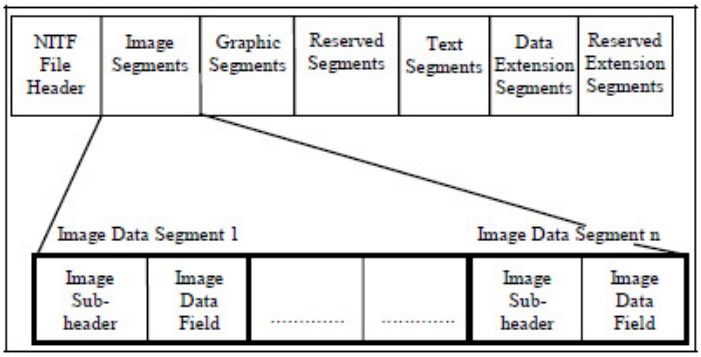


Figure 2.2: NATO secondary image format structure.

One must also note that it is very common that NSIF contains text only data and no imagery. It is very useful in a various range of missions where imagery cannot be shared.

Within the CSD implementation of the STANAG 4559, the NSIL_IMAGERY_VIEW provides imagery specific metadata attributes in addition to the core metadata set. While this view could allow a server to expose imagery in formats different from STANAG 4545/NSIF (in which case the MIME type attribute would reflect an image type different from NSIF), the server expects all imagery in this view to be in STANAG 4545 format. By introducing the 'part' structure, a new revision of the metadata model, the server now supports cataloguing NSIF files with more than one image. Each image within the file is individually described in the metadata. The metadata model also supports description of reports with the NSIF file [4].

Here are some important notes about the NSIF file and the CSD implementation of the NSIL_IMAGERY_VIEW [4] :

1. An NSIF file doesn't need to include an image segment to be a valid file according to STANAG 4545. However, to be catalogued under the NSIL_IMAGERY_VIEW it will be required that the NSIF file contains a minimum of one image segment.

2. In the case of a streaming image the NSIL_IMAGERY entity is optional in the NSIL_IMAGERY_VIEW;

3. In the case of a file-product the NSIL_IMAGERY entity is mandatory;

### 2.2.3 NATO Ground Target Indicator Format STANAG 4607

The aim of the NATO Ground Moving Target Indicator Format (GMTIF), STANAG 4607, is to promote interoperability for the exchange of ground moving target indicator radar data among NATO ISR Systems. Note that STANAG 4607 interprets the term *ground moving target indicator* to mean *targets on the surface of the earth, to include terrestrial, littoral, and deep water areas, stationary rotators, and targets flying at low speeds close to the surface of the earth* [2].

The STANAG 4607 defines the data format for ground moving target indicator radar data, regardless of the level of sophistication of the radar system and provides data that can be interpreted by any compliant ground system.

In addition to its use as a stand-alone format, the GMTI data can also be formatted in accordance with this standard and then encapsulated in either of the NATO image formats (the NATO Secondary or Primary Imagery Formats, STANAGs 4545 or 7023, respectively). This feature allows additional data, not included in this format, to be transmitted in conjunction with the GMTI data [2]. GMTI information can also be disseminated through the NATO Standard ISR Library Interface (STANAG 4559).

Each server is responsible for receiving the incoming MTI data, storing it, and making it available to the users as STANAG 4607 data via the NSIL Interfaces and the Communications Network. The key point is that the servers, the networks, and the user applications must conform to the STANAG 4559 interoperable interface requirements [2], which is the case for the CSD implementation used in this project.

In the CSD, the NSIL_GMTI_VIEW provides the specific metadata attributes in addition to the core metadata set. The file type is application/x-cgmti. Files holding GMTIF data shall be binary according to STANAG 4607 [4].

### 2.2.4 NATO Motion Imagery Format

The primary objective of the NATO Motion Imagery (MI) standard (STANAG 4609) is to provide common methods for exchange of MI across systems within and among NATO nations [3].

Conformance with the STANAG 4609 will allow any compliant system to decode all compressed data types (Standard Definition, Enhanced Definition, and High Definition) up to a minimum level but each Nation may choose to ORIGINATE one, two or all data types [3].

It also contain engineering guideline to facilitate integration of motion imagery products into the STANAG 4559 data model. However, it seems that is strongly US based and it could be assumed that future revisions of the STANAG 4609 will have corrected this [4].

Note also that it does not define a generic metadata set, but rather provides an example of a metadata set, the metadata set used for Predator generated video. It is assumed that a generic (non-Predator specific) metadata set will be developed, resulting in an updated and new mapping at some point [4].

In the CSD, the supported video files are :

1. mp2t
2. mpeg

Motion imagery is stored in the CSD using the NSIL_VIDEO_VIEW data model. Specific information available as metadata for the NSIL_VIDEO_VIEW are :

1. Average Bit Rate;
2. Category;
3. Encoding Scheme;
4. Frame Rate;
5. Number of Rows;
6. Number of Columns;
7. Metadata Encoding Scheme;
8. MISM Level;
9. Scanning Mode.

### 2.2.5 NATO Tactical Data - Link16 format

The Link 16 interface is intended to provide improved information distribution, relative navigation, and identification capability in support of inter- and intra-Allied tactical command and control and mission execution functions. The purpose of STANAG 5616 is to specify the rules, protocols, and translations required between the J series and M series messages.

J and M messages are messages that exchange digital information among airborne, land-based, and ship-board tactical data systems. It is the primary means to exchange data such as radar tracking information beyond line of sight. Link11, for M series messages, can be used on either high frequency or ultrahigh frequency. Link 11 relies on a single platform to report positional information on sensor detections [37].

On the other hand, Link16, for J series messages, was designed as an improved data link used to exchange near real-time information. It is a communication, navigation, and identification system that supports information exchange between tactical Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance (C4ISR) systems. It provides secure, jam-resistant voice and digital data exchange [41]. All systems that forward data must adhere to STANAG 5616 [1].

Within the CSD, Tactical Data Link data can be stored using the NSIL_TDL_VIEW data model. The supported file to be associated must be an *application/x-nact-link16* file type, which is a modified Link 16 data made available as binary data files with the addition of the NACT header [4].

Specific information available as metadata for the NSIL_TDL_VIEW are [4]:

1. Activity : A number that together with the 'platform' number defines the identity of a track;
2. Message Number : The Link 16 J Series message number;
3. Platform : A number that together with the 'activity' number defines the identity of a track;
4. Track Number : Link 16 J Series track number for the track found in the product. The track number shall be in the decoded 5-character format (e.g. EK627).

Since TDLs are networks on the battlefield that carry all manner of real-time messages, this category encompasses diverse types such as Participant Position Location and Identification (PPLI) and tracking management messages. Because this Layer is not as mature, this Layer is likely to change in future editions of the STANAG 4559 [4].

### 2.2.6 Other important NSIL data model

This section presents other types of data supported in the CSD within the their respective data model.

#### 2.2.6.1 NSIL Message View

The NSIL_MESSAGE_VIEW is included in the CSD implementation to store and retrieve information relative to chat, email and any other form of communication. It is not meant to store any ISR related information product.

#### 2.2.6.2 NSIL Report View

The NSIL_REPORT_VIEW is included to enable the server to better support cataloguing of exploitation products, where exploitation products are defined as products that have been generated manually, semi automatically or automatically by processing a raw sensor product and enriching it with information. Within the NATO there are several standardized report types available depending on the tasking, the types of sensors and the information needs [4]. These are :

1. Information Quality Report (IQREP);
2. Intelligence Report (ISRSPOTREP);
3. Motion Imagery Exploitation Report (MIEXREP);
4. Moving Target Indicator Exploitation Report (MTIEXREP);
5. Reconnaissance Exploitation Report (RECCEXREP);
6. WL Exploitation Report (WLEXREP).

This view will be used later on to store and retrieve encoded AIS message from the CSD.

#### 2.2.6.3 NSIL Collection Coordination and Information Requirements Management

The Collection Coordination and Information Requirements Management (CCIRM) Layer allows the Library implementing this STANAG to support some aspects of the CCIRM process. This Layer, along with the Reporting Layer and Associations in the Minimum Layer allow :

1) A CCIRM system to use the Library to provide tasking to sensor systems and exploitation systems; 2) Sensor and exploitation systems to provide the results of the tasking back to the CCIRM system.

This view may return four different and mutually exclusive products, a Collection Exploitation Plan (CXP), an Intelligence Requirement, a Request For Information (RFI), a TASK and a Geographic Area Of Interest (AOI). In the case of Geographic AOI being returned there will be no "part specific" parts. It is assumed that the RFI and the Geographic AOI are stored as XML files accessible through the NSIL_FILE.productURL. The Intelligence Requirement is an artificial metadata-only product with no specialized attributes, it is only there as placeholder to allow associations to be linked to a uniquely identified Intelligence Requirement entry. For details about an Intelligence Requirement the client will have to look at the content of a CXP file. The TASK is a metadata only product [4].

## 2.3 Coalition Shared Data Server, NPR and SC2PS Installation

Two instances of CSD were installed during the initial testing phase of this project. One was installed under Windows 7 and the other was installed on a Windows XP operating system.

In order to install all the software correctly furnished as GFI (CSD, SC2PS and NPR), it is necessary to perform the following steps :

1. Make sure that all the important updates from Windows are installed
2. Create C:\call11 directory and copy content from previous installations.
3. Create C:\Temp if it does not exist already
4. Correct the GPW template file NPR_v6.sql and make sure that at line 234, the type of TrackID "smallint" is changed for "int"
5. Install Microsoft SQL server (SQLEXPRWT_x64_ENU.exe, the free version called Express, located in Common/.../Call-up_11/software).
   (a) Select New Installation
   (b) Accept licence and next
   (c) Next
   (d) Default Instance + next
   (e) Next (don't click on "Use the same account...")
   (f) Default (Windows Auth. mode) + next
   (g) Next

(h) IMPORTANT: authorize remote access to the database:

i. In MSSQL Studio, left panel, right-click on server, select Properties → connections and make sure that the checkbox "Allow remote" is selected.

ii. Using the MS SQL Server Configuration Manager (separate tool in the start menu), Select SQL Server Network Configuration, Select Protocols, Select TCP/IP properties, and select "enable". IMPORTANT: Restart the service using the same tool (in the main panel). If there is an issue, verify the Firewall to allow the connection to the Database port. Sometimes, there is a popup from the firewall detecting the connection attempt and requesting permission, in that case, click "Allow".

6. Using the MS SQL Studio tool, connect to the SQL server (it usually uses the laptop name)

   (a) On the left side, click on the SQL server, click on databases with a right-click
   (b) Select Create a new database
   (c) Give the name CSD20 and click Ok
   (d) Repeat the process for creating the NPR database
   (e) Open the (corrected) file NPR_v6.sql (within Studio), make sure that the database in the drop down menu is not "master" but "GPW", and click "Execute!"

7. In the command prompt with administrator privileges, go to the C:\call11\CSD20\MsSqlSpatial directory and run the command:

   (a) msscmd.exe -deploy -server="NAME_OF_THE_LAPTOP" -db="CSD20"
   (b) msscmd.exe -deploy -server="NAME_OF_THE_LAPTOP" -db="GPW"
   (c) The above commands installed the geospatial functionalities in both databases, it is the equivalent of PostGIS.

8. Install Java 1.6 (32bit NOT 64bit)(jdk-6u45-windows-i586.exe, located in Common/.../Call-up_11/software). CSD won't work if you use Java 7 64bit. All defaut settings.

9. If you are skilled enough, remove the auto updated of Java.

10. Make sure that the environment variables (Control Panel → System → Advanced ... → Env Variables) are set:

    (a) JAVA_HOME = C:\Program Files (x86)\Java\jdk...
    (b) Path = .....;C:\Program Files (x86)\Java\jdk..\bin

11. The provided CSD MSI installation exec file has a bug in it, it requires a D: drive. Insert a USB Flash drive in the laptop to simulate that D: drive. This USB drive is also necessary for deinstalling the CSD server (from the Configuration Panel → Uninstall Program)

12. Open a Command Prompt window with Administrator Privileges (using the right click on the Command Prompt icon)

13. In the command prompt, run the following command (adjust according to your settings): msiexec.exe /I "C:\call11\cd1\SOURCES\CSD.msi" /quiet /L*V "C:\Temp\CSD.log" INSTALLDIRC_DRIVE="C:\" TARGETDIR="C:\call11" DB_LOCATION="C:\Program Files\Microsoft SQL Server\MSSQL10_50.MSSQLSERVER\MSSQL\DATA"

14. Verify that the C:\Temp\CSD.log file ended with Successful Installation statement. If not, deinstall everything (if necessary), made correction and try again.
15. If everything went well, the files nsiliserver.properties and nsiliclient.properties located in C:\Program Files (x86)\CSD20\dist\etc should be configured properly without needing any more modification. Although, it is possible that some adjustment is needed.
16. Test the CSD installation by running http://localhost:portnumber/nsili in a browser (Google Chrome is fine)
17. Using the brower, use the following link http://localhost:portnumber/nsili/diagtool/UserConfiguration.jsp
    (a) If login is required, use "admin" and "admin" for username and password.
    (b) Create a new user "nsili" with "create" and "update" privileges.
    (c) Create a new user "ArchiveUser" with only "update" privileges.
    (d) The password used for both was "nsili1"
18. Deactivate the Java Update Scheduler by using CCleaner (tools → Startup) and disable SunJavaUpdateSched
19. Install SC2PS (double click on C:\call1\cd2\SC2PS... ). Use default values (do not click recording...). Make sure that .NET version 4 is installed.
20. Remember to run SC2PS as administrator (right click on SC2PS.bat icon) or it won't work. Click on "Yes" when the system requests permission to run administrator stuff.
21. On the first SC2PS execution, the firewall may generate a popup, click on "Allow" if requested.
22. Recheck for Windows Update, installed them if necessary and reboot the system
23. Look at the "Release notes" for further information.

## 2.4 UAV Data Upload

The data furnished as GFI consist of several surveillance videos capture from a camera on board of a UAV. Each of these video have an associated XML file that described the necessary metadata at the time of loading it in the CSD. In order to load the data in one CSD, the SC2PS CSD capabilities were used. The specific steps to perform this task are described in 3.1.2.

## 2.5 Data Exchange between Coalition Shared Data Server instances

Data exchange between CSD instances is done through the synchronisation capability of a CSD.

### 2.5.1 Coalition Shared Data Servers Synchronization

The CSD supports four types of synchronization. From the *Release Notes* document:

**Regular** This type of synchronization only copies database entries from client to master CSD . When searching for the product, the products URL will display a link to the remote machine (client). The file location is on the remote machine. Following the provided URL link the product file will be made local to the master CSD .

**Migration Only** This type of synchronization will transfer database entries and make products local on the master CSD . The client CSD will still retain the copies and database entries for the synchronized products.

**Migration and Product Deletion** This type of synchronization will transfer database entries and make products local on the master CSD . Once the product is fetched it will perform validation that the product exists on the master CSD. If validation is correct it will proceed to DELETE the products (database entries and files) on the remote client. In order to DELETE, the user has to provide username and password with a updator valid role.

**Delete Products and Database** This type of synchronization will delete database entries and product files on the client CSD without any transfer. The product will be permanently lost on the client CSD .

In other words, there are two models of synchronisation: regular and migration. In the first case, only the metadata is transferred, the actual file (if any) is kept on the original CSD . In the second case, migration, both the metadata and the file are transferred.

#### 2.5.1.1 Synchronisation setup

In this section, we provide instructions on how to setup a synchronisation. Our instructions are mostly the same as those found within the *Release Notes* document, but with additional comments originating from our experience in setting-up an actual synchronisation between two CSD instances.

Useful tips can be found in the live application by placing the mouse over a field and waiting for a tooltip to show.

Some definitions first. In the *Release Note* document, references to master and slave CSDs are made. The difference between both is briefly given in the description of the *regular* synchronisation type given above. It is: *This type of synchronization only copies database entries from client to master CSD*. From this we conclude that a client is the host from which files will be sent to the master. Additionally, we understand that the master is also the CSD on which a synchronisation is setup. Thus:

**Master** : the CSD on which a synchronisation is set-up. The CSD that will *receive* new products.

**Client** : the CSD from which products are *taken* (copied from and possibly deleted).

In the following, the instructions given should be performed on the master CSD, the one to receive products from the client.

#### 2.5.1.1.1 Remote endpoint of the client

The first step is to identify the client, the remote CSD from which product will be taken.

Enter the URL and IOR location in the *Connection Point* box. For a Canadian CSD the location of the IOR file will be at: `http://remote_address:portnumber/nsili/csd/ior`

Where *remote address* is the IP address or domain name of the client CSD. The port number is fixed for Canadian CSDs and can be found in the CSD documentation.

For other client CSDs, the administrator must acquire the information from the remote CSD administrator.

The user will then press the *Get Library Id* button. If the client CSD address information is correct and the client CSD is accessible the client CSD library name is inserted in the *Subscribed CSD Server Library ID* box.

The release notes state it is recommended the user enter the acquired library name into the *Original Data Source Library ID* thereby implying this is an optional step. It is not. You must enter the same information found in the field *Subscribed CSD Server Library ID*, i.e. the fields *Original Data Source Library ID* and *Subscribed CSD Server Library ID* should be the same. That might not be the original intention, since entering a non-existent library id will result in no synchronisation taking place and entering a valid library id that is not the same as the subscribed one will result in a blank page being served when validating the synchronisation later (when pressing add). According to the release notes the original library id is there to help prevent circular synchronization of data.

#### 2.5.1.1.2 Time interval and type of synchronisation

The user may choose a specific date time to filter synchronization products. If no entry is provided the default time is the time of the initialization of the synchronization.

The user will then select archiving type (regular, migration, migration and product deletion or delete only). The user will need to provide username and password for migration and product deletion, and deletion only archiving types.

#### 2.5.1.1.3 Starting the synchronisation

The user will then press the *Add* button and the synchronization properties will be added to the local CSD synchronization file. If the configuration is correct the user can press the *Restart Synchronization* button to establish synchronization with the client CSD.

To remove an ongoing synchronization, select the CSD library to be removed, (check the remove check box) and press the *Remove* button. The user needs to press the *Restart Synchronization* button to definitely remove the library ID from the local CSD synchronization file.

#### 2.5.1.1.4 Status

The status of an ongoing synchronisation can be seen by clicking on the *Status* entry of the top-left menu bar, then selecting *Synchronisation* at the bottom of the next page. The status shows which synchronisations are taking place and how many products have been transferred over different periods of time.

### 2.5.2 Slowdown Mechanism

Task 5 in the Call-up described a software implementation of a data exchange mechanism intended to mimic bandwidth restriction (throttling) between two CSD instances. The restriction should go from complete stoppage to none at all.

The goal presumably is to exercise the capability of a CSD to adjust to increased timeouts. According to the *Release Notes* document, the following option needs to be modified (in the runtime script of Tomcat `C:\ProgramFiles(x86)\CSD20\dist\bin\run_tomcat.bat`) when experiencing problems related to a network slowdown:

```
-Dcom.sun.corba.transport.ORBTCPReadTimeouts=500:30000:5000:20
```

where the numbers are in this order:

1. initial time to wait in milliseconds,
2. max time to wait in milliseconds,
3. max time to wait for General Inter-ORB Protocol (GIOP) header in milliseconds, where the GIOP is the abstract protocol by which object request brokers communicate,
4. exponential backoff as a percentage when retrying.

These numbers should compensate for decreased bandwidth and latency.

Generally speaking, any slowdown mechanisms can be classified in two categories. Those operating in kernel space and those operating in user space. In the following sections we explore both and describe their pluses and minuses.

#### 2.5.2.1 Kernel space

All networking functions are implemented in kernel space, which refers to the space of memory reserved for running the privileged kernel, kernel extensions and most device drivers. The networking stack is part of the kernel space and one cannot easily insert itself there to change its fundamental behavior. Tools must be used to manipulate settings in the kernel. These tools modify parameters affecting the behavior of networking functions. Traffic control is one of those functions and we can distinguish the following:

**shaping** the rate of transmission which is bandwidth throttling but also smoothing out bursts in traffic.

**scheduling** to improve interactivity, including reordering packets.

**policing** where control is exercised on incoming traffic.

**dropping** traffic exceeding a set bandwidth is simply dropped.

Thus one can achieve a high degree of customization by carefully modifying the kernel network parameters. User-friendly tools to modify these parameters seem to be available for MS Windows Server editions. We have not found any evidence of their availability in standard editions though. Linux however comes with these tools by default.

#### 2.5.2.2 User space

User space is where most end-user applications reside. Most prominent in this category of applications are the proxies. Those are processes listening for incoming connection on a given port and forwarding any requests to its destination. Sitting in the middle, an HTTP proxy, for example, can perform all kinds of operations like denying access to certain web sites, caching often requested web pages and most important here throttling bandwidth. These tools operate entirely in user space and do not modify any underlying networking parameters.

There are tools that are not proxies and available under MS Windows that seem to operate partially in user space. The most popular are NetBalancer[1] and NetLimiter[2]. They throttle bandwidth of one process at a time, allowing to precisely fine-tune any slowdown.

#### 2.5.2.3 Advantages and disadvantages of both approaches

How would someone best implement a network slowdown? So far we have identified three solutions:

1. Modify networking parameters of one host on the network through which traffic passes.
2. Use a tool like NetBalancer or NetLimiter.
3. Use a proxy

The first solution is an elegant one because of its generality. All traffic will be throttled. Not just HTTP in the case of an HTTP proxy for example. Also, and perhaps more importantly, it mimics a real slowdown in the best way: at the network layer. However, these parameters are not easy to fine-tune. Their modification is best left to a dedicated throttling tool.

The second solution certainly makes use of such dedicated tools. NetBalancer for example, will throttle all traffic to and from a particular process. We have experimented with NetLimiter 3 Pro

[1] `http://seriousbit.com/netbalancer/`
[2] `http://www.netlimiter.com/`

under Windows 7 and it works as advertised allowing one to slow down to 1 byte per second any process.

The third solution is a viable one as long as the protocol being proxied is HTTP and as long as the process being slowed down allows a change of configuration to point to a proxy first. Unfortunately, it does not appear the CSD is configurable in such a way.

Since it's unclear how two CSDs communicate and since we do not know if they'll continue communicating the same way, we are proposing solution 1 above as the best one: modify kernel parameter to mimic a slowdown at the network layer. In order to best mimic a real slowdown it should be done en route, on the wire. We mean by that a slowdown implemented by a middle-man sitting on the wire between the two machines under tests, the ones hosting the master and slave CSDs. And the best device for this is a router. We are using a router with a modified, open-source firmware optimized for traffic control.

The main advantages for adopting such an approach is the generality of the solution. All traffic can easily be throttled, not just a specific protocol. This also allows one to throttle an entire subnet, all computers behind the router can be throttled at the same time, very efficiently. Throttling can go as low as 1 kbps, that is 1024 bits per second or 128 bytes per second. The interface to change this parameter is a standard web page easily accessible from any machine. The one drawback however is that both CSDs need to be on different computers.

If that cannot be the case, we are suggesting using NetLimiter to slowdown the Apache process responsible for running the CSD code (see figure 2.6).

#### 2.5.2.4 The router

The router provided is a Linksys E2500 with a modified firmware, TomatoUSB. The E2500 was chosen precisely because the open source TomatoUSB firmware can be installed on it. As most inexpensive routers today, it provide a wireless interface. However, for obvious security reasons we have disabled this interface.

TomatoUSB is an alternative Linux-based firmware for routers with Broadcom chipsets. There are different *flavors* of TomatoUSB each made and maintained by different individuals. The one we chose is by Shibby[3], probably the most popular and user-friendly version.

TomatoUSB provides advanced Quality of Service (QoS) with 10 unique QoS classes defined, real-time graphs display, prioritized traffic with traffic class details and client bandwidth control via QoS classes. One can throttle a single IP address to an entire subnet, down to 1 kbps, all through a simple web interface.

##### 2.5.2.4.1 Initial configuration of the router

We are providing the router already configured for bandwidth limiting. Here we describe the steps we took to do so. Access the router at 192.168.1.1 when connected on one of its LAN ports

[3] `http://tomato.groov.pl/`

(the four blue ports). Default username and password were kept, it is *admin* in both cases. We recommend changing those as soon as possible.

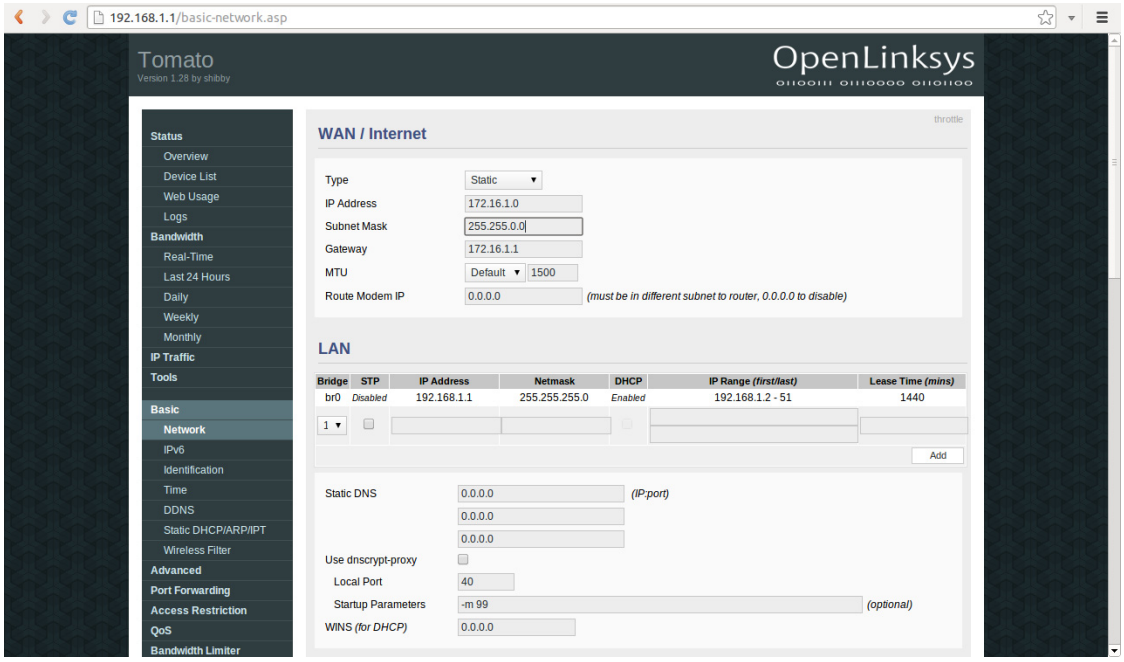


Figure 2.3: WAN interface configuration

The WAN interface configuration is shown in figure 2.3.

In Basic → Network (left navigation bar), you'll find a section WAN/Internet. Under that section, modify the following parameters:

1. Choose *Static* from the *Type* dropdown menu.
2. Set the IP address of the router to 172.16.1.0
3. Set the subnet mask to 255.255.0.0
4. Set the gateway to 172.16.1.1

Details for enabling QoS are in figure 2.4.

In QoS → Basic Settings (left navigation bar), under section Basic Settings, check *Enable QoS*. Under section Outbound Rates/Limits, write 80000 in *Max Bandwidth Limit*.

The limiting Bandwidth interface is shown in figure 2.5.

In Bandwidth Limiter (left navigation bar), under section Bandwidth Limiter for Lan (br0), check *Enable Limiter*. Write 80000 in both fields *Max Available Download* and *Max Available Upload*.

The initial configuration for limiting bandwidth is now done.

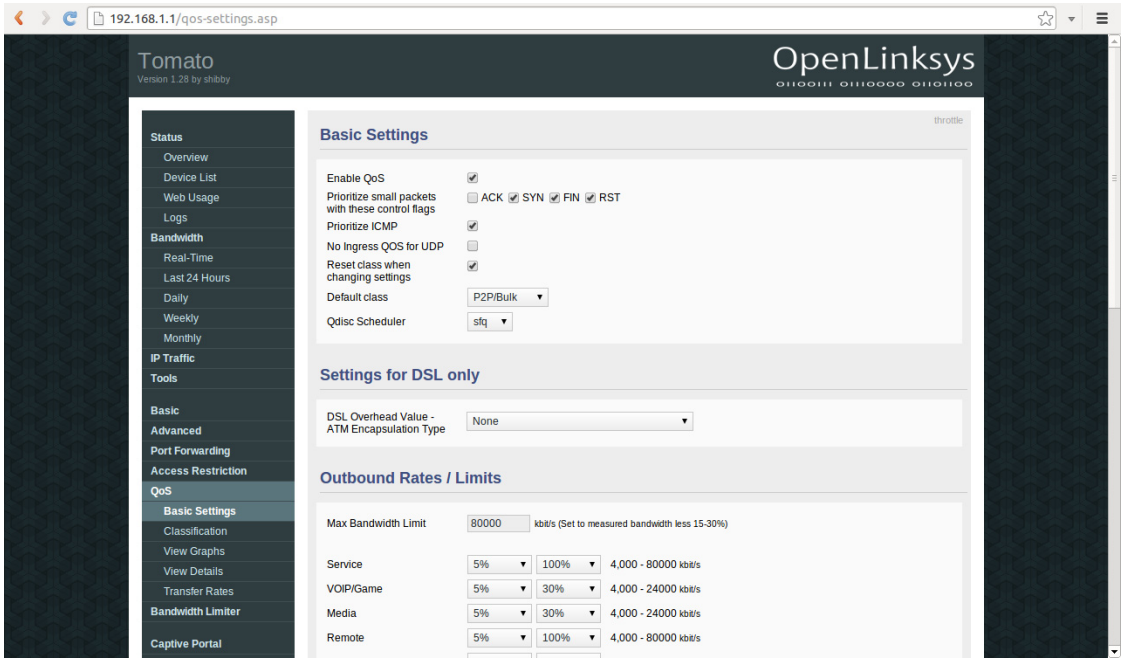


Figure 2.4: QoS configuration

#### 2.5.2.4.2 Limiting bandwidth with the router

From the Bandwidth Limiter page (left navigation bar), one can find a table under section Bandwidth Limiter for Lan (br0). This table has the following fields:

**IP | IP Range | MAC Address** You can enter an IP address, or a range of those or a single MAC address to which the following limits will be applied.

**DLRate** The target download rate, from 1 kilobit per second up to 99,999 kilobits per second. This rate is what we are trying to achieve. It might be a bit lower or a bit higher.

**DLCeil** The absolute maximum download rate. Above that, packets are dropped.

**ULRate** The target upload rate, from 1 kilobit per second up to 99,999 kilobits per second. This rate is what we are trying to achieve. It might be a bit lower or a bit higher.

**ULCeil** The absolute maximum upload rate. Above that, packets are dropped.

**Priority** The priority of the traffic coming from this source, i.e. whether or not it should be processed faster than other sources.

**TCP Limit** Maximum number of active connections for this source.

**UDP Limit** Maximum UDP connections that can be opened per second.

We suggest the following default value for all limiting rates:

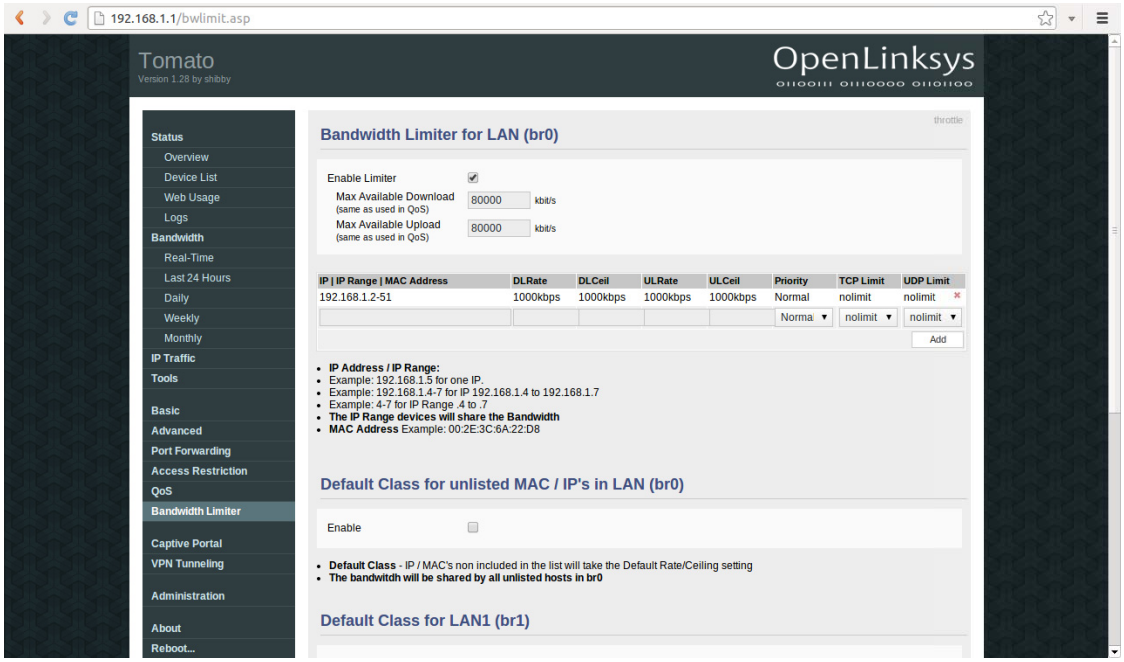


Figure 2.5: Limiting bandwidth

- Enter a range of IP addresses: 192.168.2-51. All IPs in this range will be bandwidth limited.
- Keep a *Normal* priority.
- Keep *nolimit* for both TCP Limit and UDP Limit.

To limit rate for all connected machines, simply enter the same desired rate in the fields DLRate, DLCeil, ULRate and ULCeil as in figure 2.5. You can go as low as 1 kbps or as high as 99,999 kbps. One kilobit is 0.125 kilobytes or 125 bytes.

It is important to understand that bandwidth limiting occurs at the WAN-LAN interface. Machines on the same subnet, those within the 192.168.1.* subnet will not be limited. In practical terms, this means that one of the CSDs (we suggest the one representing the UAV, the client CSD) be connected to the WAN port, the yellow one. The other CSD, the master which will receive data from the client, should be connected to one of the four LAN ports, the blue ones.

#### 2.5.2.5 NetLimiter

We found NetLimiter superior in use over NetBalancer. NetLimiter is available at `http://www.netlimiter.com/`. Version 3 is necessary to run under Windows 7.

We ran into issues while testing NetLimiter 2 under Windows XP at very low rates of 500 bytes/sec. No such problems were experienced under Windows 7 or at higher rates.

#### 2.5.2.5.1 Usage

A CSD runs under the Apache Web Server. It is that process that must be slowed down. The name of that process is rather unfortunate, it is Commons Daemon Service Runner (see figure 2.6). Rate limiting a process is as simple as checking the boxes under DL Limit and UL Limit and double clicking the numbers beside each checkboxes to adjust the rates.

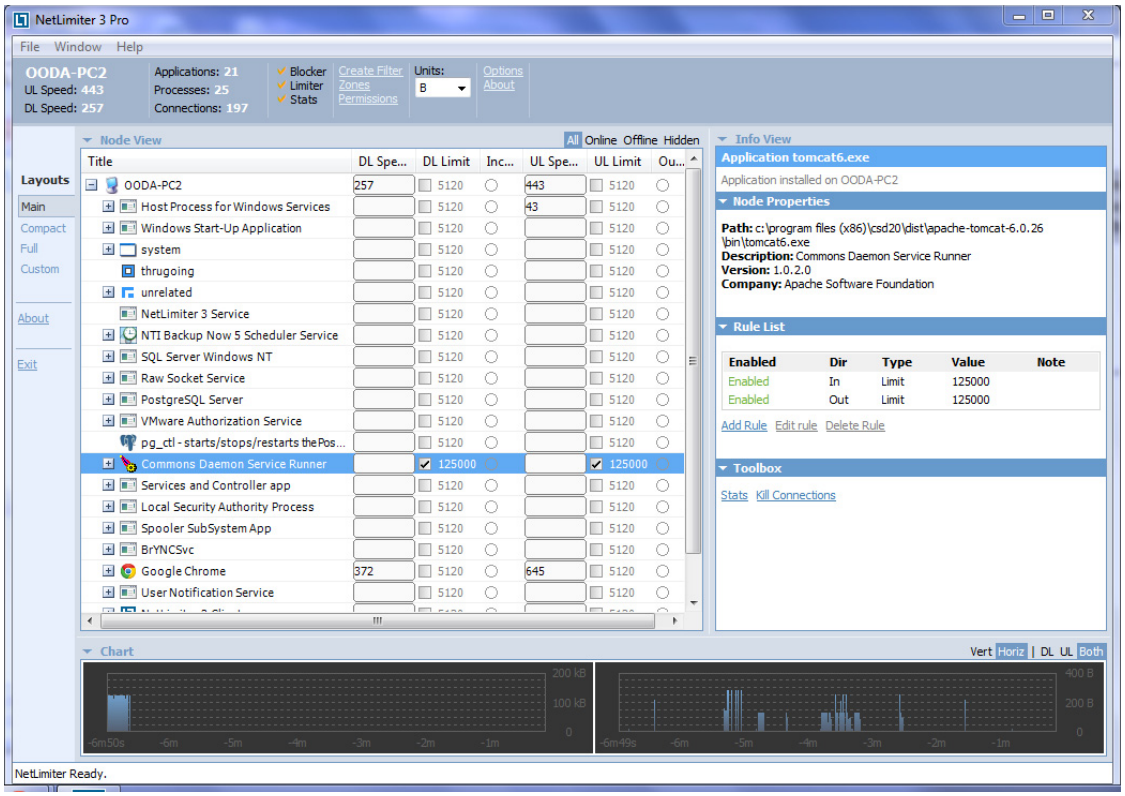


Figure 2.6: NetLimiter 3 Pro

## 2.6 Data Exchange Behavior in a network of Coalition Shared Data Server instances

This section presents the results and several observations related to the use of CSD instances in a network configuration. The proposed scenarios, described in sections 2.6.1 and 2.6.2, were not entirely tested as the CSD design prohibited some configurations and proved some assumptions wrong. Nonetheless, several interesting observations were made and are presented here. These observations will highlight some behavior of the CSD which are not very well documented and will make the reader more aware of how to synchronize CSD instances when they are placed inside a complex network.

The initial state of the network is represented by figure 2.7. CSD A is the Master CSD and

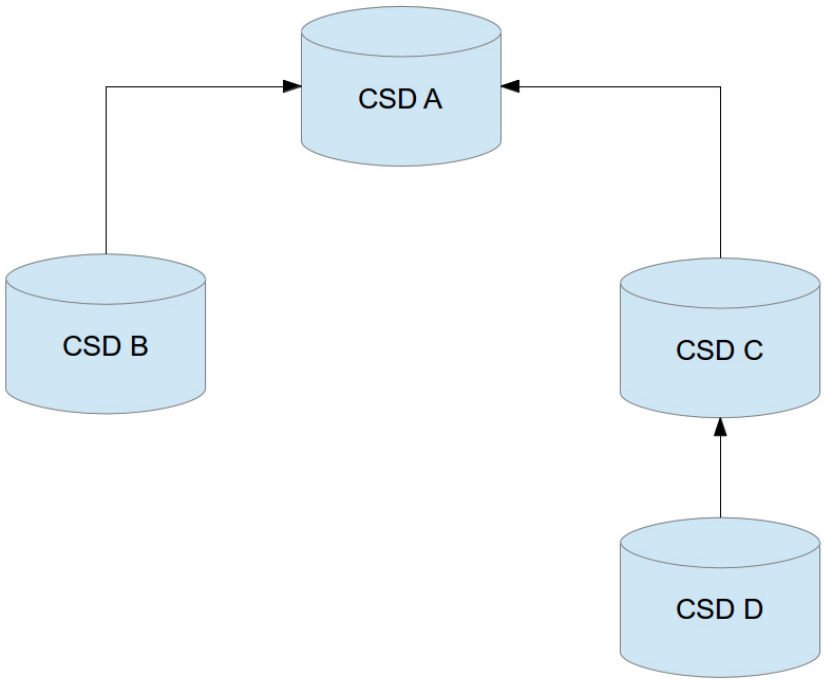


Figure 2.7: Initial state of CSD network

synchronize the product libraries created with CSD B and C. CSD C synchronize the product library created with CSD D.

### 2.6.1 Scenario 1 : Connection failure before resynchronization

In the first tested scenario, a connection failure occurs between CSD A and CSD B and C. CSD A is therefore isolated from the other CSD instances. Before the connection drops, all CSD instances were synchronized.

The steps realized to perform this scenario are :

- A network reconfiguration is made such that CSD C becomes the Master CSD, and CSD B is now connected to C. Products are added to CSD B and D.
  - Different products are added to both instances of CSD.
  - The same product is added to both CSD instances.
- Wait until synchronization is over.

- Re-establish connection with A so that the network is at its initial state.
- Reconfiguration such that CSD A becomes the Master CSD, and CSD B and C are now connected to A (i.e remove the connection between B and C)
- Synchronize and observe the behavior of the products.

Some important observations were made during the testing of the scenario 1. First, the synchronization cannot be made blindly and this is a very important CSD concept. Lets consider the three instances of CSD (A, C and D) from the initial configuration. Configuring CSD A to synchronize with CSD C will not automatically retrieve all the products created on CSD D even if those products are locally present in CSD C. It is not the entire product content of the CSD instances which is synchronized, it is the different products libraries available and each instance can contain multiple libraries. Its is important to know that each product created on particular CSD instance is part of a particular library linked to that particular CSD instance. Therefore, all products created in the CSD instance D are part of, lets say, library D. Products created in CSD instance C are part of library C, and so on.

As an example, in order for CSD A to synchronize with products libraries C and D through CSD instance C, two synchronization configurations must be defined (assuming that CSD instance C is synchronizing with the library D from CSD instance D):

- Retrieve library C products from CSD C;
- Retrieve library D products from CSD C.

And if CSD instance B would be connected to CSD instance C instead of being connected to CSD instance A, we would have to set up another synchronization on CSD A to retrieve its products library :

- Retrieve library B products from CSD C.

There is no way to configure the synchronization mechanism to retrieve all the libraries at one time. This implies that if one wants to retrieve products from a particular library, he must be aware of the library existence and the location where this library is synchronized from.

It also means that CSD instances are particularly well suited for a bottom-up data exploitation where CSD instances in the middle of this bottom-up process produce enhanced products from lower-level library. Therefore, only enriched information products reach the upper-level where more decisions are likely to be taken. The possibility of including a reference to a lower level products also enable the retrieval of the original product from all the different layers.

Another important CSD concept is that the data products inside a CSD library can never be deleted. Once a product is inserted in a CSD, there is no way it can be removed. This is made per design. The only way to remove certain products from the result set of a particular query is to mark them as obsolete. An obsolete product will never come up in a product search and its file URL will be removed from the metadata.

This scenario was made to answer some questions related to our understanding of the CSD behavior. These questions along with our observations that relate to them are listed below.

- Is there a duplicate happening in the CSD A due to reconfiguration of the network?

  No, there is no duplicate happening in A. The reason is pretty simple. Each product in a library has a particular Unique IDentifier (UID). Therefore, at synchronization time, only products with new UID or updated products are synchronized. It is believe that a check is made on both the UID and the product update time during CSD libraries synchronization.

- Is there a duplicate happening in the CSD C due to reconfiguration of the network, and if so, is the same product replicated 3 times in CSD A when the connection is re-established?

  There is in this case a duplicate, but the product was not replicated 3 times. This is normal as they are not exactly the same product. They share the same content but do not have exactly the same creation date and are not part of the same products library and therefore have different UIDs. The CSD have no mechanism to explore the actual content of the product.

- Is the order in which the synchronization starts ( B to A before C to A) have an impact on the synchronization process and outcome?

  The order have no impact on the final state of the CSD instance A.

- What are the final states of each CSD instance at the end in terms of products content?

  The content of CSD A and C is exactly the same.

- What is the difference on the synchronization process (migration versus regular) at the end on the content of the CSDs (products, metadata, etc. . . )?

  No difference on the end content of the CSDs with the exception that the products are remote and not local in the regular mode of synchronization.

### 2.6.2 Scenario 2 : Connection failure during synchronization

In the second tested scenario, a connection failure was planned to occur between CSD A and CSD B and C during synchronization. CSD A would therefore be isolated from the other CSD instances. At connection drop, CSD instances were still under synchronization process. CSD B and C would have been sending products to CSD A.

The network would have been reconfigured such that CSD C becomes the Master CSD, and CSD B is now connected to C and time would have been allowed such that all synchronizations would have been completed between B and C. Connection would then be re-establish between A and both B and C.

This scenario was not possible. Both CSD B and C are not aware of the existence of CSD A. As such, both CSD instances sending their data to the instance A is not possible. CSD A control the synchronization sequence and synchronize libraries in the order in which the administrator entered them in the synchronization configuration.

Another scenario was therefore designed to test if a product that was being sent to instance A and to instance C would be taken again in A from C and would therefore create a duplicate entry. A looping in the CSD synchronization was attempted. It was interesting to see that such a configuration for synchronization is forbidden by design. As an example, lets consider the following :

- CSD A is configured to synchronize with library B through CSD B;
- CSD C is configured to synchronize with library B through CSD B.

If the administrator add the following synchronization configuration :

- CSD A to synchronize with library B through CSD C.

The CSD instance A will automatically drop the synchronization with library B through CSD B. Only one synchronization configuration per library per CSD is possible.

It is impossible to duplicate a library even when passing through several different CSD instances.

This scenario, even if not completed as intented, was made to answer some questions related to our understanding of the the CSD behavior. These questions along with our observations that relate to them are listed below.

- Verify that the product that was being sent to A has been sent to C. Is there a duplicate of said product in CSD A when connection is re-established?

  This question is no longer valid. However, no duplicate of a product ever occurs.

- If there is no duplicate, it could mean :

  that C was not able to send its product as it was already existing and CSD A is waiting for CSD B to send it. Or C sent it and CSD A is managing this process all by itself.

  The CSD synchronization mechanism is managing this process all by itself.

- What happens if B never reconnects? Will the product that was being transfer from CSD B will be accepted from CSD C or is it lost forever as CSD A is waiting for CSD B to reconnects to finish its synchronization of the product?

  A product is never lost, at least we did not observe this behavior.

This page is intentionally left blank.

# Part 3

# CSD Data Visualization

This section presents three different options for the visualization of UAV data contained in the CSD.

The interaction with the CSD was tested using :

1. Sensor Command and Control Planning Suite (SC2PS) : a proprietary software labelled with the Control Goods restriction;
2. NASA World Wind : an open-source Java-based Software Development Kit;
3. Quantum GIS : a open-source Geographical Information System where interface components can be developed using the C++ QT library and plugins can be added using the Python programming language.

It is important to note that the development made under NASA World Wind and QGIS are exploratory and were done in a very limited amount of time. They show the potential of each library and different options to interact with the CSD. More complete interface and interactivity could be developed if one would devote more time in development.

## 3.1 Visualization using SC2PS

The SC2PS is a real-time, multi-sensor application to exploit data from soldier systems, ground-based sensor sources, tactical aerostats, and UAVs. SC2PS gives commanders a powerful tool for analysis, mission planning, and decision making. SC2PS data is both Multilateral Interoperability Program (MIP) compliant and CSD compatible.

The SC2PS was installed on both Windows XP and Windows 7 operating systems. However, there is some problem under Windows 7 due to permission on writing and updating directories. The lack of permission to write and update in the directory Runtime of SC2PS and all the directories beneath is breaking many functionalities. However, SC2PS does not throw any error message with the exception of an inaccessible mission.dat file when trying to create an Event marker when watching a video previously loaded.

Giving the user all the permissions on the Runtime directory and all the sub-directories will unlock all the functionalities of SC2PS.

### 3.1.1 Connect to a CSD with SC2PS

From the *CSD Configuration Menu*, the analyst must specify the following parameters :

1. User name
2. Password
3. Location of the CSD
4. Location of the CSD CORBA Initial Object Reference (IOR)
5. Time for wait
6. Number of connection attempts before failure

Once these parameters are correctly entered, simply press the *Connect* button.

### 3.1.2 Upload Motion Imagery from UAV with SC2PS

The upload of the UAV video data into the CSD was made using the SC2PS. In order to load motion imagery within the CSD, one must perform the following steps :

1. Click on the *CSD* menu;
2. Click on the *CSD upload* menu item
3. A window will appear, make sure the *Auto-load XML* is checked
4. Press the *Add* button and select the file you want to upload, note that they must be compatible with the standard defined in the STANAG 4545, 4607 or 4609
5. Click the *Open* button
6. Click the *Upload* button.

### 3.1.3 Query the CSD and visualize Motion Imagery with SC2PS

In order to query the CSD to visualize the motion and still imagery from the UAV, one must perform the following steps :

1. Open the CSD query window from the SC2PS *CSD* menu.
2. In the *CSD Search Criteria* panel, build a query that looks for all entries that are videos (or images) by ticking the *of type = VIDEO* (or *of type = IMAGERY*) box. Run the query using the *Search* button on the top right of the window.

3. In the *Query Results* panel, all the found entries will be listed. There are 72 in our case. The thumbnail column however always displays a red $X$ image for every entry.

Other metadata can be used as search criteria such as, but not limited to :

1. Creator of the file;
2. Title of file;
3. Geographic coverage and time window coverage;
4. Etc...

In order to view the data that have been returned as query results, the following steps are necessary :

1. Go in the query result tab (tabs are shown at the bottom of the query form);
2. In the *Query Results* panel tick one or more boxes in the *Selected* column to select which entries to download.
3. Click the *Download* button to download the selected entries.
4. A pop-up window appears asking *Download all products referenced by selected products?*, click yes. A new table will appear in the *Product Download* panel. Under the *Download Status* column, the status of the selected download briefly changes to *Queued* then the download starts.

### 3.1.4 Exporting video / imagery exploitation product to the CSD

Video and imagery exploitation products can be created in SC2PS and exported to the CSD for archiving. This would be a standard way to, for example, tag a particular frame in a video.

In order to do so, follow these steps :

1. Open a video :
   (a) Click on the *Video* menu;
   (b) Click on the *Open video* menu item, a window will appear;
   (c) Click on the button with the directory icon;
   (d) Select your video and click *Open*.
2. Press the *Play* button if the video does not start automatically
3. In order to tag a particular frame of the video :
   (a) Press on the button with MILSTD symbol;
   (b) A new element will appear in the *Contacts* directory of the *Mission* folder;
   (c) Right-click on the newly created contact;

(d) Click on the *Create Intelligence Report* or on the *Create IExploitation Report*, override/-modify/fill the metadata.
(e) Press the *OK* button.
(f) A new report will appear in the *Reports* folder within the *Mission* folder.

You can export the contacts and reports to the CSD by right-clicking on them and clicking on the *Export to CSD* contextual menu item.

## 3.2 Visualization using NASA World Wind

*NASA World Wind* is an open-source virtual globe developed by NASA and the open source community for use on personal computers [42].It is a Java-based Sofware development Kit, enabling the development of custom interface for geospatial data display and interaction.

World Wind can be expanded by using one of many add-ons - small extensions that add new functionality to the program. Possible types of add-ons [42]:

1. Point layers - simple XML files displaying placemarks (point of interest) as icons
2. Trail layers - paths (routes, boundaries)
3. Line features - XML with a list of points visualized as a line or wall
4. Polygon features - XML with a list of points visualized as a filled polygon (flat or extruded)
5. Model features - XML used to load 3D textured meshes
6. Place names - specific points (such as cities, hills and buildings) that are assigned text labels Image layers - high resolution imagery for various places in the world
7. Scripts - files that control camera movement

This SDK provides all the necessary tools to develop a custom interface to visualize the products from the CSD.

As an example of query and visualization, an interface to interact with the CSD has been developed with the API provided in the SDK to showcase some of its capabilities. This interface is presented in figure 3.1.

The *CSD* menu allows users to :

1. Connect to a CSD instance via a configuration window (represented in figure 3.2);
2. Query the CSD instance (queries are hardcoded for the moment retrieving all AIS and video available);
3. Upload AIS contacts to the CSD via a file dialog window
4. Set the selected products as obsolete (Reports only);
5. Exit the program.

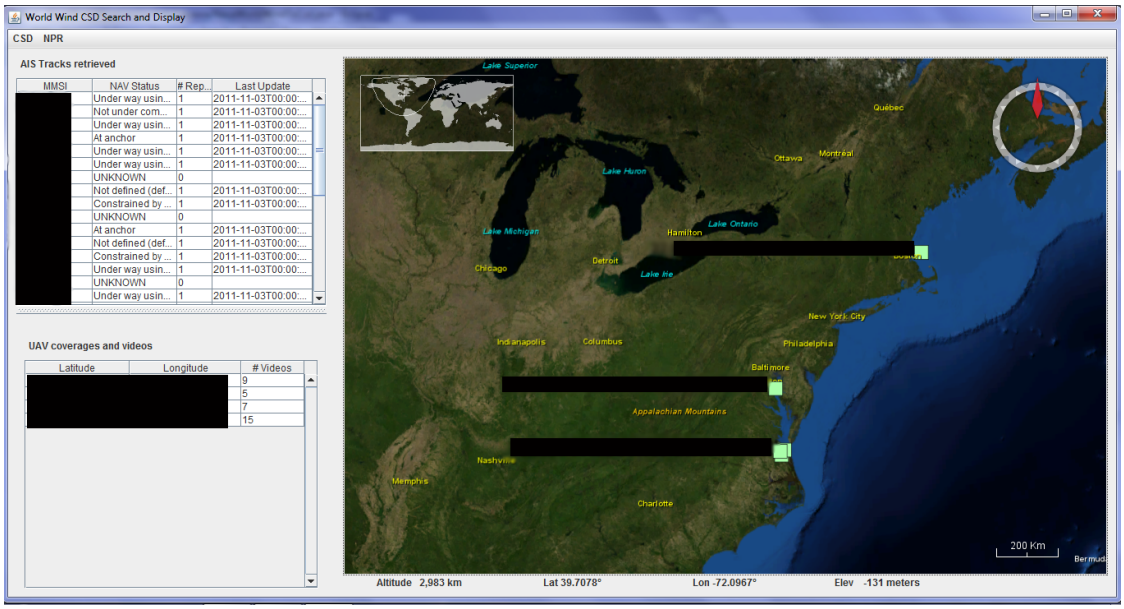


Figure 3.1: NWW interface with AIS contacts

The *NPR* menu allows users to transfer Automatic Identification System (AIS) contacts from the CSD to an NPR instance.

In the developed prototype, the AIS contacts are grouped into AIS tracks internally based on their Maritime Mobile Service Identity (MMSI). All constructed tracks are displayed in the Track Table on the upper left on the interface. The AIS tracks are displayed on the virtual globe using a red dot. They could be displayed using the MILSTD2525-A symbology available in the World Wind SDK, however it would require more time to learn how to control the displayed information over this symbology as the documentation is minimal. Right-clicking on the track symbol display a pop-up window with track related information (Name, MMSI, Callsign, International Maritime Organization (IMO) number, Destination, etc...).

The UAV videos, on the other hand, are regrouped under the UAV coverage they are part of. The available coverages and their approximate positions are listed in the UAV coverage Table on the lower left of the interface. These coverages are represented by yellow rectangles on the virtual globe. Right-clicking on them provides the list of available videos (see figure 3.3). Clicking on a video will trigger the download of the file and the video will automatically open and play in Windows Media Player (see figure 3.4).

Clicking on an item on the Track or UAV Coverage tables will zoom at the item position on the 3D globe.

The NASA WorldWind SDK enable the development of custom visualization and interaction software. It already provides the developer with a flexible framework. In addition, this framework could be extended to suit particular requirements that are not yet supported by this SDK.

The main drawback of using the NASA WorldWind is that it is usually required for defence

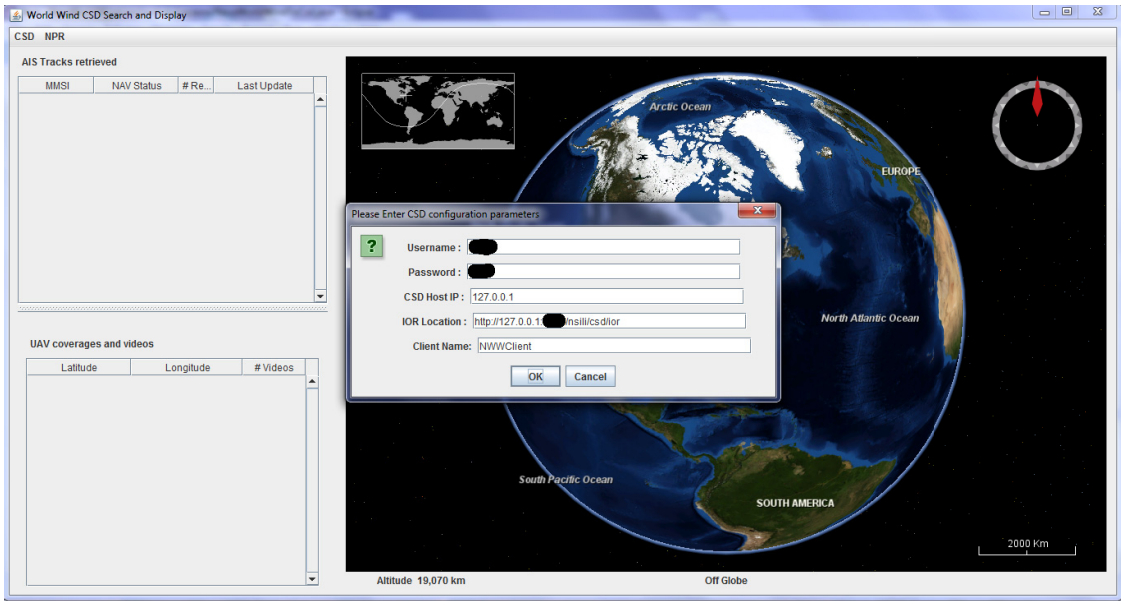


Figure 3.2: CSD connection configuration window

application to work offline. This requirement necessitate some change in the SDK configuration and package structure. Also, the image overlay of the 3D globe are downloaded in real time while using the SDK. Therefore, it is necessary browse the globe prior to putting the computer offline in order to have image overlay when working with no internet connection.

To summarize, the NASA WorldWind SDK:

1. Provide a complete Java-based SDK enabling the development of custom software and interface;
2. Functionalities for the CSD are not clouded by any other interface component;
3. Can be integrated directly with the CSD API.
4. Can work offline with minor modifications;
5. Have moderate documentation and support but is relatively easy to use.

## 3.3 Visualization using Quantum GIS

QGIS is a free and open-source geographic information system. QGIS is an official project of the Open Source Geospatial Foundation (OSGeo). There are numerous company offering commercial support for it. It runs on Linux, Unix, Mac OSX, Windows and Android and supports numerous vector, raster, and database formats and functionalities. It is built as a core set of functions augmented by a plugin system.

A screenshot of QGIS loaded with multiple layers is shown below in figure 3.5.

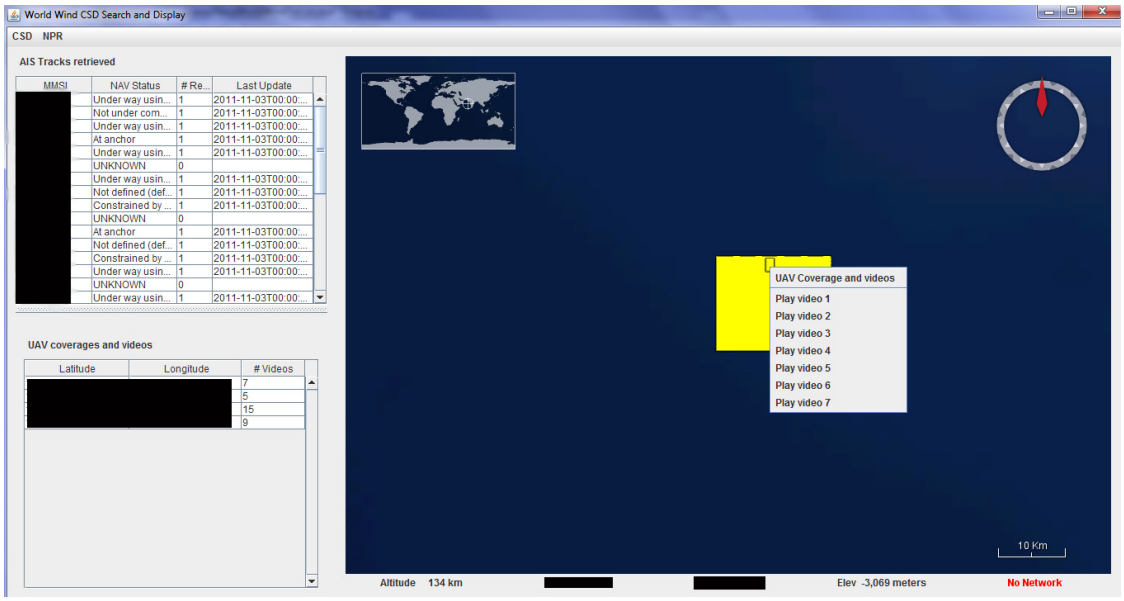


Figure 3.3: UAV coverage and video list

### 3.3.1 The CSD plugin

The plugin interfaces with the CSD through the web interface by making HTTP requests, specifically POST requests. Product listings are retrieved by making an appropriate request and parsing HTML pages. There is no need to use the Java API directly. This choice of using the web interface is motivated by the fact that QGIS plugins are written in Python and re-using an existing interface, the web one, is more efficient.

Reports are shown as MMSI labeled dots on the map. Videos are not displayed due to a lack of development time.

The plugin interface provides fields to input the CSD hostname and port. Shown in figure 3.6 is the CSD plugin interface.

The plugin will fetch products found in the map area currently displayed. A screenshot of a part of Africa is shown in figure 3.7 with reports found in that area.

### 3.3.2 Pros and cons of QGIS

QGIS is a tool for mapmakers. Its main purpose is to manipulate different layers of features like points, polygons, lines, etc. to gradually build a map. The final product is an image in a common format like gif or png. QGIS is not an interactive tool for purely visualizing maps, it's use to build them rather.

This has consequences on the use-case at hand, namely showing CSD products on a map. First, it is slow at rendering a map. Whenever a new portion of a map is to be shown, e.g. by zooming or


Figure 3.4: Video playing in Windows Media Player

panning, the rendering time is quite long. It's not interactive. Second, QGIS feels too cumbersome to use for such a simple task. The intended use here is to zoom, pan and get at each report's attributes, but these simple tasks are hampered by the slew of features offered by QGIS.

#### 3.3.2.1 Developing the plugin

Here we report on the plugin development. The plugin architecture is based around a set of Python wrappers around the C++ API. Python calls are translated directly to C++. Because of this, no documentation exists for the Python API. The only documentation available is the C++ one and it is very scarce. The direct consequence of this is a very long and convoluted development time.

This is slightly offset by the fact that Python is a very expressive language greatly reducing any development time.

## 3.4 Remarks on Visualization

This section presents some remarks on the visualization development made under QGIS and NASA World Wind.

### 3.4.1 Interaction with the CSD

Two different approaches were tested :

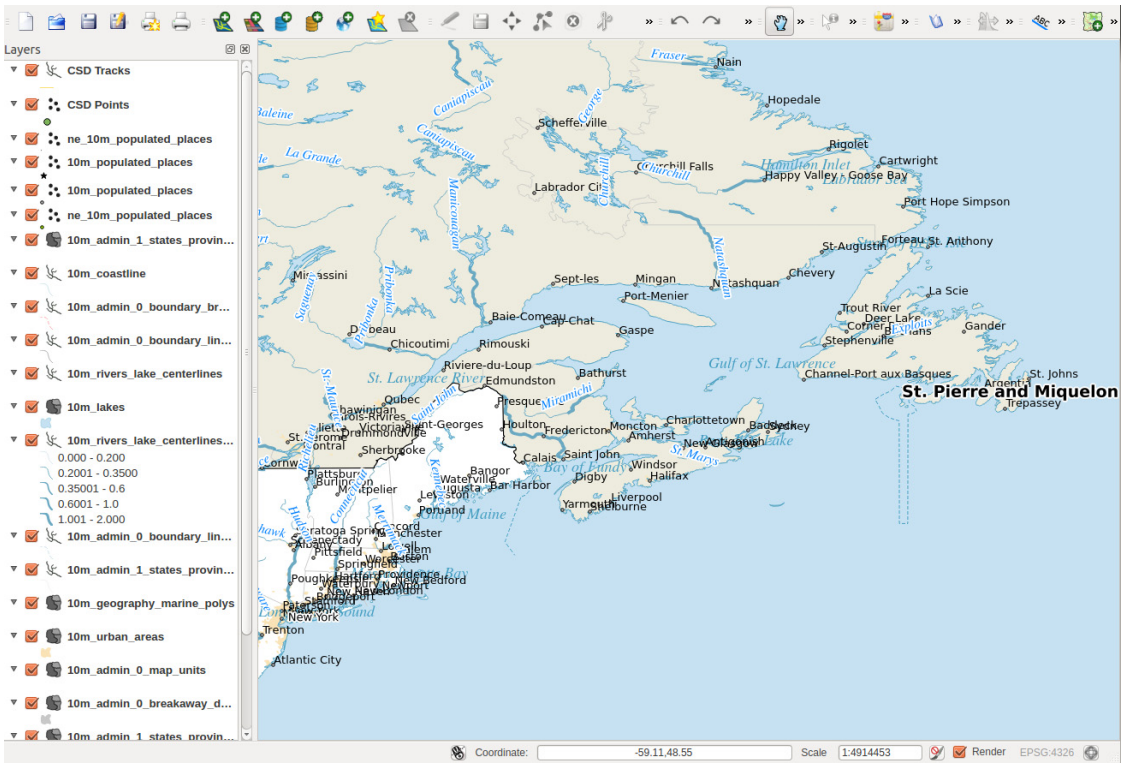


Figure 3.5: QGIS with multiple layers shown

1. Using the Java API to connect and query the CSD with NASA World Wind;
2. Using the web service to query the CSD and retrieve the results with a Python code base.

It is worthwhile to mention the complete opposite could have been done (wrap the Java API and call it from Python and use the web service from a Java code base). This remarks only illustrates that various approaches can be made to connect and query the CSD, depending on a project requirements or programmer preferences.

### 3.4.2 Extracting reports from the CSD

Each reports must be extracted from the CSD by fetching an XML file using the web interface, i.e. making an HTTP call. That is one HTTP call by product. This is an I/O bound operation and is very time consuming. There are ways to reduce this time by making asynchronous calls, but we observed only a three-fold reduction in time. Great but not good enough for an interactive visualization. Unfortunately, aside from making calls directly to a Java API function (whitout using a HTTP GET) there are no faster way of extracting products from the CSD.

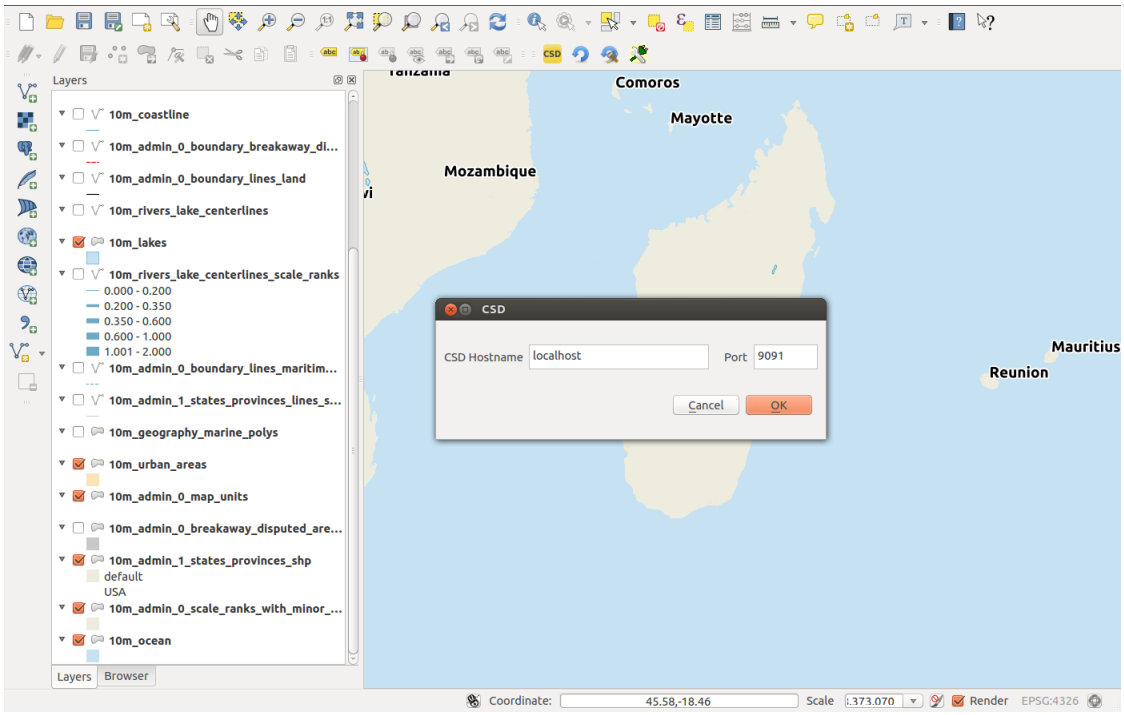


Figure 3.6: CSD plugin

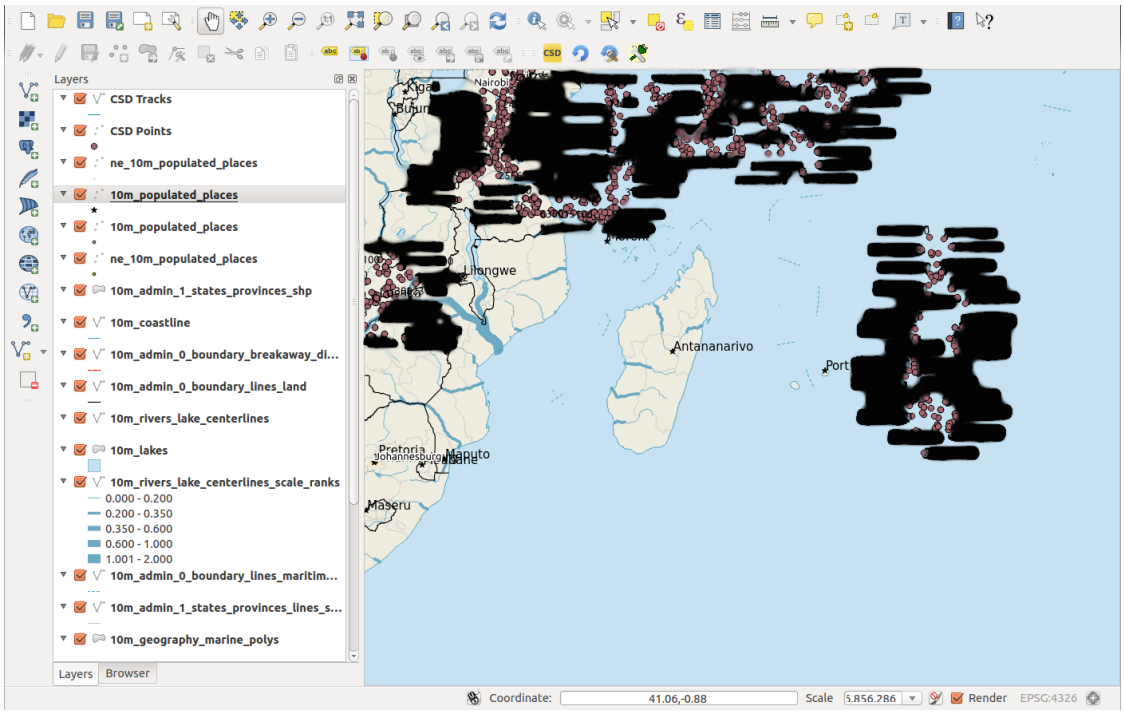


Figure 3.7: CSD reports shown

# Part 4

# Coalition Shared Data Server and the Global Positioning Warehouse

This section describes the work performed with regards to GPW v.2.0 or NPR. It details some conceptual differences between the structure of the CSD data model and the NPR database structure.

## 4.1 Naval Position Repository Installation

The NPR instance was installed on a Windows 7 OS using the SQL script that was received as GFI. No particular steps was taken in the realisation of this task. No bug was detected and the installation went smoothly.

## 4.2 CSD and NPR data structures comparison

Some problems in the NPR database strucutre are preventing the storage of UAV video data to be transfered in a NPR context. These are the following :

- UAV video data does not contain any identifiers of the observed entity. Therefore, any UAV data sent to NPR would have to contain an UNKNOWN entity ID.

  NPR's purpose, as we understand it, is to keep track of the position of known entities (ships mostly). Therefore, it might not make much sense, from the point of view of the NPR framework, to insert a lot of data that is not associated to any entity within the database. This is a problem of principle: Should the NPR database be filled with data that doesn't fit its intended design?

- While NPR allows unknown as an entity identifier, there are indications within the GFI that all contacts reporting an unknown entity essentially point to the same single entity entry - while they do in fact represent different entities.

- Transferring images and videos from CSD to NPR could be difficult. They could theoretically be added to the GPW database as a binary blob but might also require the addition of new tables since the current ones only allow *varchar* data.

  An alternate solution would be for the NPR database to contain only a web-link to a repository location of the video or image. A CSD user would then have to follow the provided link and download the media from the remote location. This is an easy implementation as far as NPR is concerned but it shifts the problem to one of availability and accessibility of the added repository.

- As UAV data contains a coverage area stored as a polygon, an additional geometry table would be required within NPR, for its storage. The new table could be used as a generic geometry metadata holder for contact reports. In other words, it should accept any object of type geometry so that it may serve the purpose of storing other geometry types.

In conclusion, the current state of the NPR makes it impossible to store video data from a UAV. The lack of identity in the said video is the main problem. Also, the NPR allows *varchar* data only. If the NPR retains the attachment information for the old GPW, this should be investigated as a possible solution once the entity ID problem have been solved.

## 4.3 AIS and the CSD

This section describes the work that was performed in relation to the decoding of AIS messages and finding a solution to feed them to the CSD in order to retrieve them later on and store them in the NPR.

### 4.3.1 AIS Decoder Implementation

The implemented AIS decoder is essentially the same one used for Maritime Situational Awareness Research Infrastructure (MSARI) [34]. The decoder can handle all message types including messages that employ tag blocks. The Project Authority (PA) has indicated however that only messages 1 to 3 and 5 were of interest and as such, the mapping of data from AIS to XML was implemented only for this subset of message types.

### 4.3.2 NIEM XML exchange standard

The AIS data are sent to the CSD encoded, meaning that one must decode them when they retrieve it from the CSD. Under this call up, we built AIS decoders that decode the AIS message and represent it under the NIEM XML exchange standard. The NIEM is an XML-based information exchange framework from the United States.

NIEM represents a collaborative partnership of agencies and organizations across all levels of government (federal, state, tribal, and local) and with private industry. The purpose of this partnership is to effectively and efficiently share critical information at key decision points throughout the

whole of the justice, public safety, emergency and disaster management, intelligence, and homeland security enterprise. NIEM is designed to develop, disseminate, and support enterprise-wide information exchange standards and processes that will enable jurisdictions to automate information sharing [38].

The NIEM is divided in several communities. The one of interest for AIS dissemination is the Maritime domain community. This community supports Maritime domain awareness, i.e the effective understanding of anything associated with the global maritime domain.

The schema description of the NIEM for the maritime domain can be downloaded from the National Information Exchange Model web page `https://www.niem.gov/technical/Pages/version-3.aspx`.

Using the .xsd files, one can build Java classes in order to integrate the standard exchange model in its java code. However, it is not trivial. In order to create Java files the following steps are necessary :

1. Download and unzip JAXB RI 2.2.7 from `https://jaxb.java.net/2.2.7/`
2. Create a bindings.xjb file to resolve name conflicts in the different schema. The bindings.xjb file used to generate the Java code for this project is copied in the annexes of this document.
3. At the command line, under Linux, type the following : ./bin/xjc.sh -nv -extension -cp lib/jaxb-impl.jar -b ./bindings.xjb -d ../generated path-to-niem/niem/domains/maritime/3.0/maritime.xsd. If you are using Windows, use xjc.bat instead of xjc.sh
4. Compile the Java classes and create the jar file. Make sure that all the jar files in the lib directory of the JAXB package are first in the classpath as the JRE contains a faulty JAXB jar file.2

The niem.jar file generated for this project has been included in the deliverables.

### 4.3.3 AIS Data to CSD

The CSD as it was developed can support a wide range of ISR data. All the storage, query, access and exchange mechanism are based on different NATO standards (STANAGs).

Currently, there is no accepted and implemented STANAG for AIS data. However, a work around is possible by considering an AIS report to be some kind of exploited raw sensor data.

Using this approach, a problem still remains. The NIEM XML structure of the AIS message cannot be fitted in the currently supported standard in the CSD. We have therefore adopted a work around. The AIS message was stored in the CSD using the NSIL_REPORT_VIEW. As stated in the STANAG 4559, the NSIL_REPORT_VIEW is included to enable the server to better support cataloguing of exploitation products, where exploitation products are defined as products that have been generated manually, semi automatically or automatically by processing a raw sensor product and enriching it with information [4].

As mentioned in section 2.2.6.2, this view can refer to file products that contains pre-defined, supported report types. However, none of those report types contains all the place holder necessary to store the AIS information. Therefore, a decision was made to store the complete AIS message under its NIEM XML structure under a XML node of a ISRSPOTREP message type. The NIEM XML message is stored within the SummaryOfCollectedInformation XML field. The XML-based AIS message is enclosed within a ![CDATA[]] container in order to avoid any XML validation on it. Figure 4.1 represents the flow of the AIS messages from the encoded version of them to the NPR database.

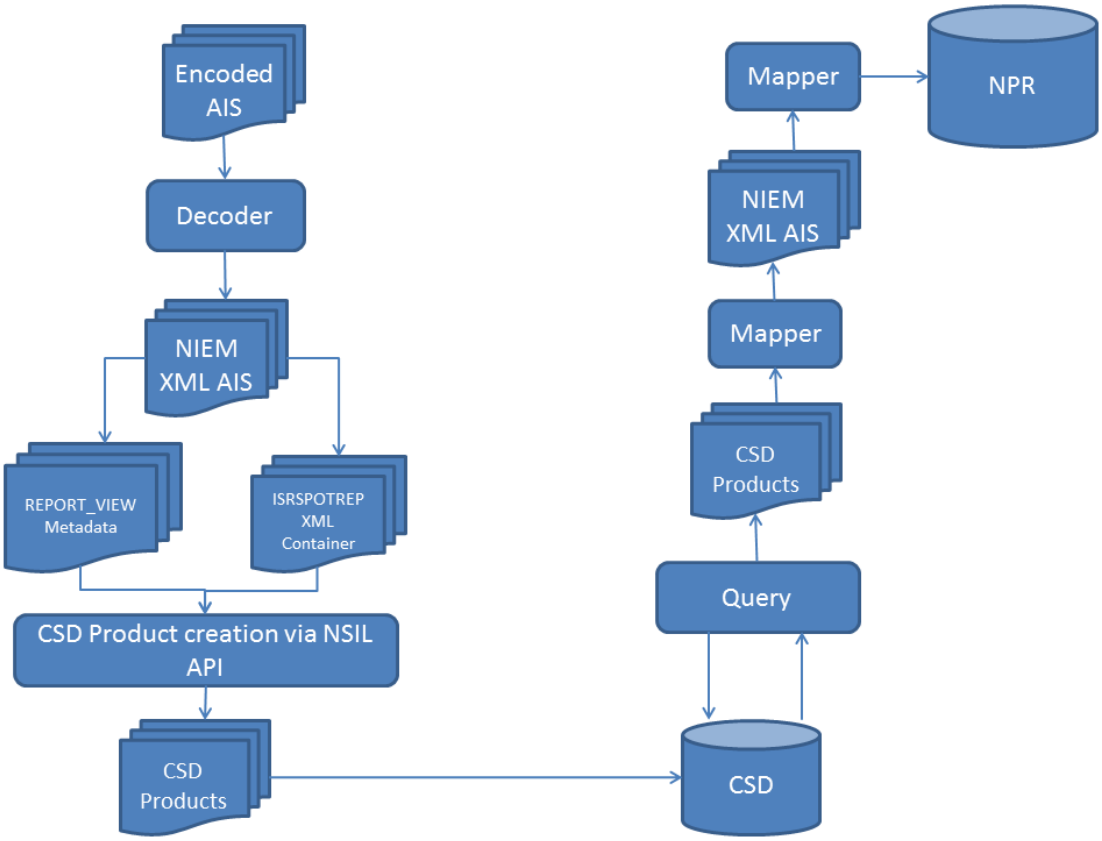


Figure 4.1: AIS messages flow

There is however a major drawback. As the AIS messaging is not yet supported by a standard and is not implemented in the CSD, SC2PS cannot retrieve and display the AIS contacts on its display. Some modifications would therefore be required. Even with another software, custom modification would be necessary, as some query mechanisms, AIS decoders and display capabilities would have to be added.

## 4.4 Data Exchange between CSD and NPR

The slowdown of the data exchange between the CSD and the NPR instance is realized using the router to limit the communication between two processes. However, in this case, a Java program is used to establish connection between the CSD and GPW as the AIS data must be queried, transformed and stored in the NPR instance.

UAV data exchanged between the CSD and NPR databases undergoes a conversion from Extensible Markup Language (XML) format to Structured Query Language (SQL) insertion queries. The correct container for UAV AIS reports of types 1-3, within the NPR database is a contact report. However, the structure of the NPR database makes it such that this transition between databases is not always straightforward.

A contact report requires a reference to a known entity, navigational status, raw data and provider. Therefore, before an AIS contact report can be persisted, the corresponding entries to the previously mentioned dependencies must either be found or generated.

The transition of AIS messages between the CSD and NPR databases has been successfully implemented for message types 1-3.

AIS messages of type 5 affect only entity data and metadata, their conversion from one database model to the next was not possible during the time frame of this call-up.

This page is intentionally left blank.

# Part 5

# Maritime Tactical Command and Control

This section presents an overview of architecture and functionalities of the GCCS-M, also known as the Defence Information Infrastructure (DII) COE, and the MTC2, which is considered to be the next generation of GCCS-M. It also presents a high level comparison of both systems. As the MTC2 documentation was not made available this study is based on information and documentation found on the internet such as MTC2 industry days presentations, technologies roadmap and progress report from US agencies and publicly available contractual information from the major US agencies involved in the MTC2 development such as Defence Information System Agency (DISA).

## 5.1 Global Command and Control System - Maritime (GCCS-M) / DII COE

In the past, most computing systems built for defense were stovepipe systems, i.e systems that operate in total isolation from the rest of the computing environment. This is in contrast to a system of systems model, where data flows seamlessly from one business process to another.

The COE concept can be best described as being [7] :

1. An architecture and approach for building interoperable systems;
2. A reference implementation containing a collection of reusable software components;
3. An infrastructure for supporting mission-area applications;
4. A set of guidelines, standards and specifications that describes how to reuse software and how to properly build new.

### 5.1.1 GCCS-M / COE Concept and Architecture

The DII COE or GCCS-M 4.x is a framework for the creation of a set of cooperating computing enterprises. Its goals included the elimination of stove-pipe systems and cost reduction via software reuse, reduced need for system administration, and simplified system integration [15].

### 5.1.2 The COE concept

The DII COE was designed to be hardware independent; it operates on a range of operating systems running under standards-based systems. DISA has developed versions of the COE for Windows NT, and for a variety of flavors of Unix, including Solaris and HP [10].

The COE must be understood as a multifaceted concept which deals with 4 specific facets ([7], [22]):

1. A system foundation;
2. An architecture;
3. A reference implementation;
4. An implementation strategy.

As a System Foundation, the COE is a foundation for building other systems. Specific COE components are selected by system designers to build their mission application. Each built system uses the same API, the same approach to integration, and the same set of tools for enforcing COE principles [9].

As an architecture, the DII COE model is analogous to the Microsoft Windows paradigm. It provides a standard environment, a set of standard off-the-shelf components called segments, and a set of programming standards that describe how to add new functionality to the environment [20]. The DII COE is designed to run on PCs and workstations; it extends the Windows paradigm, which allows applications to coexist, by providing the capability for mission applications to share data and services/functions at the server level [10].

As a reference implementation, it means that algorithms and the way the algorithms are implemented are identical between different platforms, and the only differences are in the binary code. This reference implementation is the key to ensure interoperability and software reuse ([7], [22]). It might change over time to take advantage of new technologies or fix reported problems.

As an implementation strategy, it emphasizes incremental development and fielding to reduce the time of integration of new functionality. It is a process of continually evolving the baseline to take advantage of new technologies as they mature and to introduce new capabilities([7], [22]). This strategy also considers legacy systems and how such systems may be integrated into the COE [22].

### 5.1.3 The COE architecture

The architecture of the COE is presented in Figure 5.1 (from [11]). The concept of segment is the first concept encountered by anyone when dealing with the DII COE. A segment is a unit of software or data that has been packaged such that it can be installed on a DII-compliant computer using the software installation tool of the DII COE. All software that is to be installed on a DII computer must first be put into segment format [15].

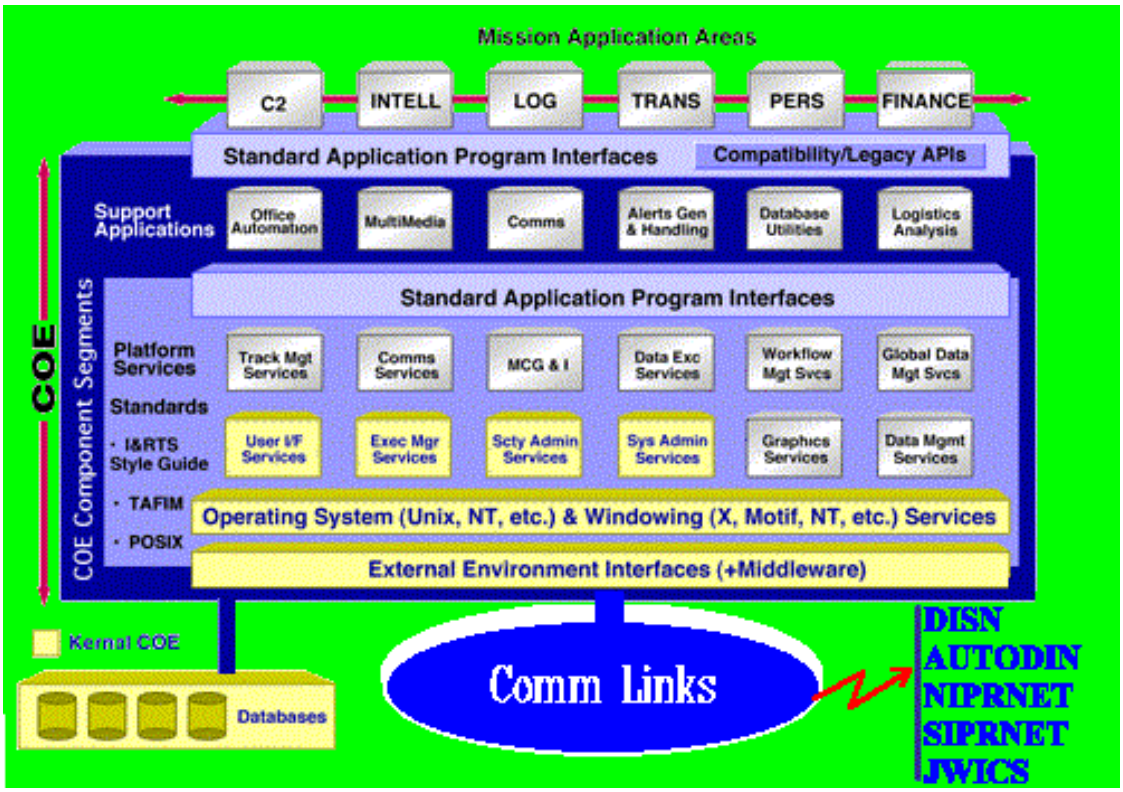


Figure 5.1: GCCS-M / COE architecture.

This concept facilitates software reuse. It forces the developer to prepare the software for integration. Once in segment format, a software package can easily be transferred between programs offices and used in a variety of contexts. Segments are defined as a collection of related functions as seen from the perspective of the end user, not the developer [9].

The software that comprises the DII COE is placed into three categories: the kernel, the common support applications, and the infrastructure services.

#### 5.1.3.1 The COE kernel

Every computer that is to be a DII computer must have the kernel installed upon it. The COE kernel is the minimal set of software required on every platform regardless of how the platform

will be used [10]. A DII COE kernel contains the operating system and windowing environment, but it will normally include several other features [9]:

1. a basic System Administration function,
2. a basic Security Administration function,
3. an Executive Manager function (e.g., a desktop GUI such as Windows NT or Common Desktop Environment),
4. a utility for creating privileged operator login accounts,
5. a utility for creating nonprivileged operator login accounts, and
6. COE tools for segment installation.

Most of these functions are provided by COTS products, and in the case of NT, may even be provided by the operating system itself. The COE kernel is used to define and establish the basic runtime environment context that an operator inherits when he logs in (among other actions, this typically means to establish which environment variables are defined and their initial setting). The kernel is intentionally designed to be as small as possible, to be as much COTS as possible, to provide a common starting point for loading segments to build up the system, and to provide an extensible but common runtime environment for segment execution. The segment concept and mission specific application capabilities are discuss in the next section [10].

#### 5.1.3.2 The COE Infrastructure services

The infrastructure services are largely independent of any particular application, They include :

1. Networking, system and security administration services;
2. Communication service to send /receive data;
3. Distributed computing services;
4. Print services;
5. Multimedia / collaborative services;
6. Web servers;
7. Data access services at the database and file management level.

The infrastructure services are services that are deemed to be of general use [15].

#### 5.1.3.3 The COE Common Support Application services

These services are much more specific to particular mission domain. They include :

1. Alert Service for routing, prioritizing and managing alert messages;

2. Correlation service which correlates information from sensors and other sources of information;
3. Mapping and cartography service to handle the display of maps and imagery;
4. Messaging service to parse and distribute military format messages;
5. Web browsers;
6. Data access services for common data access methods.

The common support applications are desktop applications that are deemed to be of general use [15].

GCCS-M, as we know it, is composed of the COE core segments and mission specific application segments.

### 5.1.4 Data access and sharing

COE enforces the concept of a system of systems, where multiple systems are able to freely and easily access and share data. This is achieved by the COE Shared Data Engineering (SHADE) component. This component includes :

1. The Data Access application part of the Common Support Applications and;
2. The Global Data Management and the Data Management services part of the Infrastucture Services.

The SHADE approach provides the engineering techniques and components which COE developers can employ to resolve data sharing problems. Use of SHADE did improved data consistency, reduce redundancy, and promote true data interoperability. SHADE does not rule out multiple copies of the same data created for performance sake, but it does manage the duplication to ensure that all databases are kept synchronized [9].

The main problem is that information is not easily shared. Communications between GCCS-M nodes are made between pre-determined point to point connections within systems and applications on disparate networks. A producer "pushes" information to predefined consumers [8]. This information push can have significant effect on the network load as some of the pushed information might not be necessary for some particular subscriber.

Some other challenges related to the COE information sharing concept are [8]:

1. User might be unaware that information exists;
2. User knows it exists but cannot access it;
3. User can access it but cannot exploit it due to a lack of understanding.

The network centric data strategy that will be presented later will address these weaknesses.

### 5.1.5 Functionalities

Also, the information output is mainly related to situational awareness (i.e Recognized Maritime Picture), which is represented as tracks and maps. It is not able to address the needs of the navy with regards to the planning and execution of missions.

Global Command and Control System (GCCS) limitations stem from the fact that each military Service found it necessary to produce a tailored version of GCCS to support individual Service missions. This has led to limitations on a Task Force Commander's ability to use common applications, share data and coordinate planning activities, drawbacks that are most obvious when joint operations are being planned and executed. As an example, GCCS-Joint baseline and Extensible Common Operating Picture (XCOP) baseline.

## 5.2 Maritime Tactical Command and Control

This section describes the architecture of the MTC2 and provides an overview of some of its core functionalities.

According to [14], the United States Navy information technology strategy is in need of change due to operational inefficiencies and interoperability problems. Ships and submarines have been forced to integrate various redundant systems in order to provide vital mission critical intelligence. The end result of these inefficiencies is as follows:

1. Loss of command and control agility and responsiveness;
2. Unacceptable timeframes to replace legacy systems;
3. Limited bandwidth per sailor;
4. Aging technology that is cost prohibitive to replace;
5. Limited reuse of technology;
6. Network security vulnerabilities.

On a high level, fundamental differences are from a paradigm shift from *need to know* to *need to share*. Also, there is the *as a Service* technology which is included in all levels of the underlying MTC2 architecture.

MTC2 will also provide apps to fill gaps at the Force Level and Maritime Operation Center. The *as a Service* technology will also greatly facilitate rapid software delivery and software reuse and therefore reduced the cost of system development and enable new technology to be tested and fielded more easily.

The MTC2 system will be based around the concept of network centric enterprise system and build on Google/Amazon-like Cloud Architectures. It will leverage the Navy's computing infrastructure provided by the Consolidated Afloat Networks and Enterprise Service (CANES). MTC2 is planning to field to the MOCs in late FY14.

### 5.2.1 Architecture of the MTC2 - The *as a Service* technology

This section presents the expected architecture of the MTC2. From a bottom up perspective, the MTC2 architecture will be made of four levels of services, namely :

1. Data as a Service;
2. Infrastructure as a Service;
3. Platform as a Service;
4. Software as a Service.

Figure 5.2 (from [13]) represents the as a Service architecture concept while Figure 5.3 (from [18]) presents this architecture in relation to the MTC2.

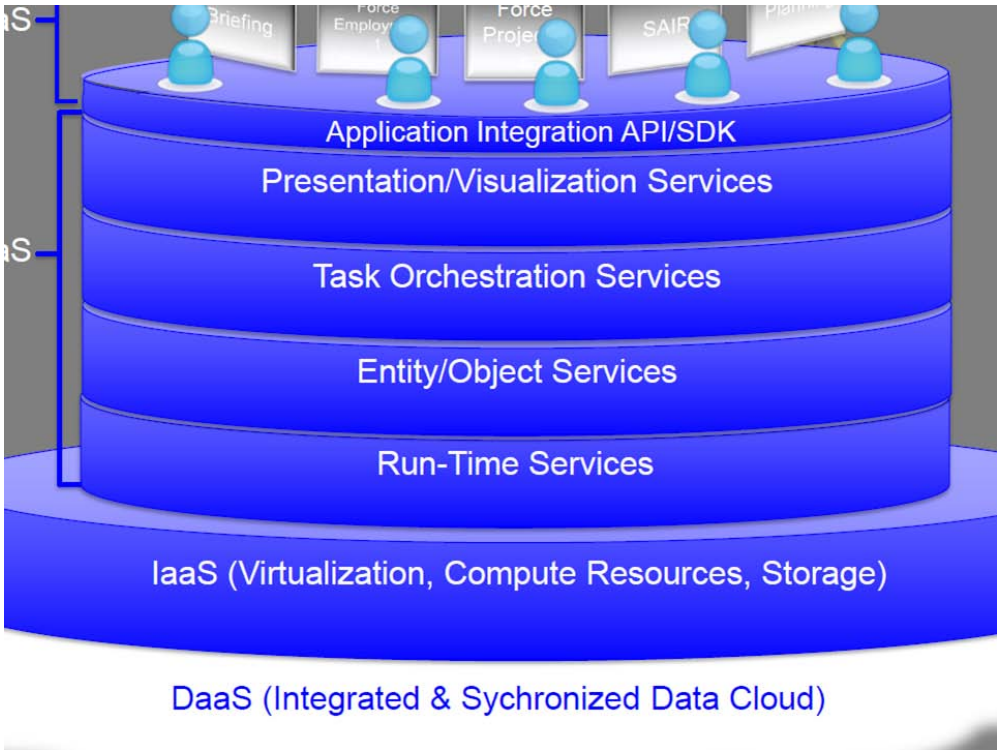


Figure 5.2: MTC2 as a Service architecture.

In addition, MTC2 is expected to be built on Google/Amazon-like cloud architectures. Therefore, as represented by figure 5.4 and figure 5.5 (both from [18]), in case of network problem, it could reorganize itself into more regional cloud temporarily while waiting to be connected to the main cloud again.

The following sections describe all of the *as a Service* layers that are part of the MTC2 architecture. Their role, the underlying technology as well their benefits and drawbacks are presented.

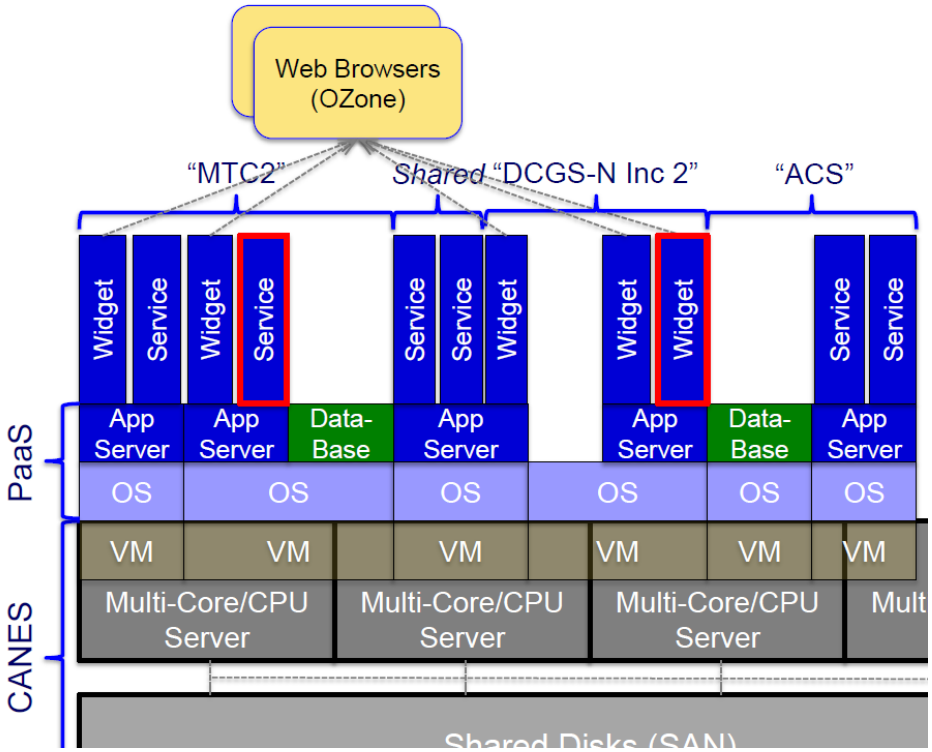


Figure 5.3: MTC2 as a Service architecture.

### 5.2.2 Data as a Service - Net-Centric Enterprise Services data strategy

Data as a Service (DaaS) is a cousin of Software as a Service (SaaS) (described later). Like all members of the *as a Service* family, DaaS is based on the concept that the product, data in this case, can be provided on demand to the user regardless of geographic or organizational separation of provider and consumer. Additionally, the emergence of service-oriented architecture (SOA) has rendered the actual platform on which the data resides also irrelevant [36].

DaaS brings the notion that data quality can happen in a centralized place, cleansing and enriching data and offering it to different systems, applications or users, irrespective of where they were in the organization or on the network. As such, DaaS solutions provide the following advantages [36]:

1. Agility - Users can move quickly due to the simplicity of the data access and the fact that they don't need extensive knowledge of the underlying data.
2. Cost-effectiveness - Providers can build the base with the data experts and outsource the presentation layer, which makes for very cost-effective user interfaces and makes change requests at the presentation layer much more feasible.
3. Data quality - Access to the data is controlled through the data services, which tends to improve data quality, as there is a single point for updates.

Interoperability is also a strategic goal of the Net-Centric Enterprise Services (NCES) Data Strat-

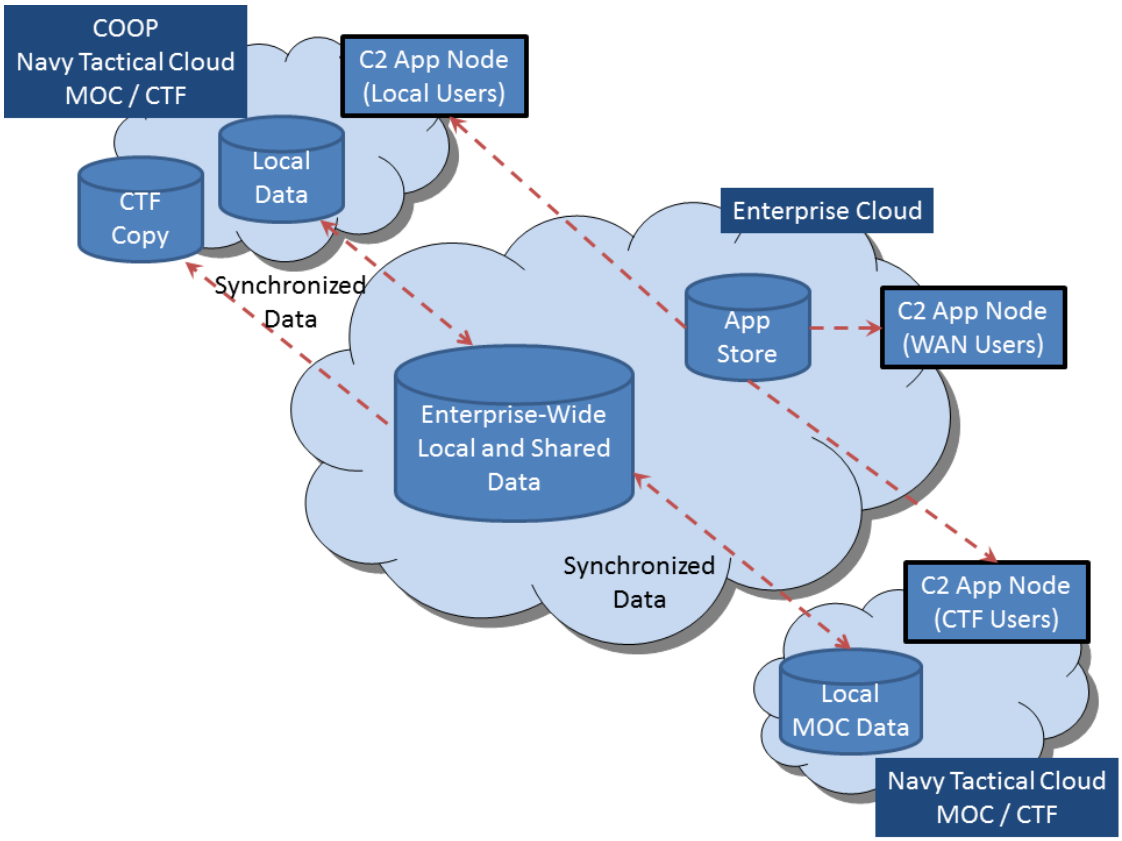


Figure 5.4: MTC2 Cloud-based CONOPS.

egy. Interoperability is the ability to share information among components while preserving its accuracy, integrity and appropriate use. Usability requires the technical condition of interoperability (mapping and matching) plus understandability through data component definitions and descriptions [27]. The data strategy of the MTC2 offers the following in support of the interoperability goal [27]:

1. Make data sets understandable by publishing associated semantic and structural metadata in the Metadata Registry or federated registry.
2. Enable mission and business processes, including their semantics and data structures, to be reused where possible and mediated where required through careful componentizing and registration of associations.
3. Decentralize data management to Community Of Interest (COI) to allow prioritization and collaboration based on immediate operational needs while providing enterprise infrastructure for self-synchronization on a larger scale.

Note that the CSD implementation of the STANAG 4559 supports the DaaS as seen by the MTC2.

The adoption of a network centric data strategy within the MTC2 resolves the problem where

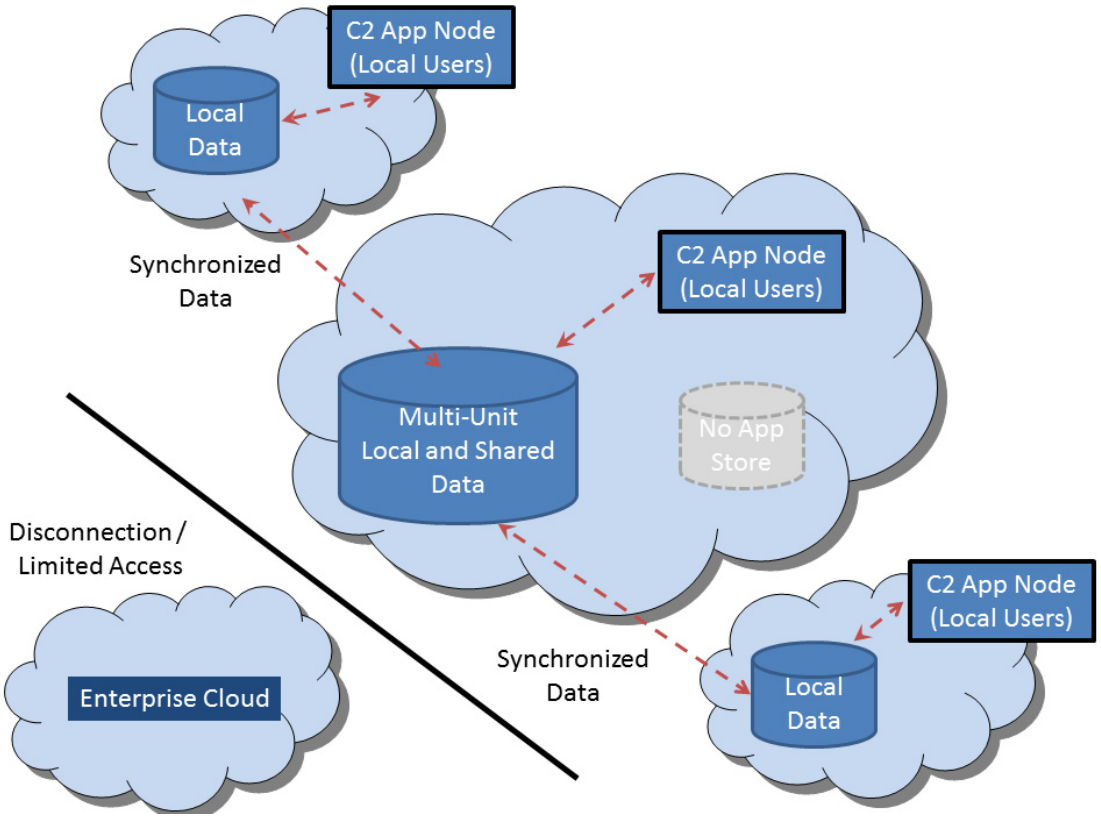


Figure 5.5: MTC2 Cloud-based CONOPS in case of DoS.

information is not easily shared across different systems. According to [8], it will enable a move from :

1. Pre-determined point to point connections within systems and applications on disparate networks;
2. Producer pushes information to predefined consumers.

to the following situation :

1. Systems and applications are web-service enabled to expose their information;
2. Authorized known and unanticipated consumers pull or subscribe to what they need regardless of who produced the information.

#### 5.2.2.1 Community of Interest

A COI is an organizational construct for working collaboratively to establish clusters of data interoperability that cross formal boundaries. COIs are important because they make explicit and

widely visible (publish names, advertisements of what they are doing, who's involved etc.) vital information sharing task groups that would not otherwise be even recognized as organizations [27].

COIs are generally responsible for publishing and managing metadata via the Metadata Registry and its federates as well as formulating and managing rules concerning usage of the net-centric capabilities they offer [27].

With regards to the goals of the NCES data strategy, both the COIs and the enterprise that provides the services have a important role to play in order to achieve them.

Table 5.1: Role of enterprise and COI towards data strategy goals

| Goal | Enterprise | COI |
|---|---|---|
| Make data visible | Maintain Discovery Metadata Specifiations (DMS) to facilitate search; develop and maintain enterprise search capability | Tag data holdings with DMS, extend for COI specific criteria; register access services in enterprise service registry |
| Make data accessible | Maintain repository of acceptable standards for web-based services; develop and maintain enterprise federated service registry | implements access services |
| Make data understandable | Maintain Metadata registry. develop and maintain federated metadata registry for semantic and structural metadata | Develop and maintain vocabularies, taxonomies for data exchange; register these agreements in metadata registry |

### 5.2.3 Infrastructure and Platform as a Service : The Consolidated Afloat Networks and Enterprise Service (CANES)

Today's environment is typified by dedicated hardware for specific applications. Information is tied to the application, the location and the operating system. This approach can lead to poor utilization of computing resources and require additional hardware and software to be purchased to accommodate dynamic usage and future growth [29].

Net-centric Computing Infrastructure will leverage distributed computing resources to provide infrastructure that appears to the end-users or applications as one virtual capability. Shared computing and data storage resources will be virtually allocated and the mechanism for doing

so will be transparent to users. By virtualizing how users view the computing infrastructure, an organization can begin reducing technical and administrative barriers to sharing resources, and provide more agile and scalable support for information sharing [29]. Net-centric computer infrastructure is usually defined as Infrastructure as a Service (IaaS).

IaaS is a provision model in which an organization outsources the equipment used to support operations, including storage, hardware, servers and networking components. The service provider owns the equipment and is responsible for housing, running and maintaining it [32].

Characteristics and components of IaaS include [32]:

1. Utility computing service.
2. Automation of administrative tasks.
3. Dynamic scaling.
4. Desktop virtualization.
5. Policy-based services.
6. Internet connectivity.

On the other hand, Platform as a Service (PaaS) is a category of cloud computing services that provides a computing platform and a solution stack as a service. In a PaaS model, the consumer creates the software using tools and/or libraries from the provider. The consumer also controls software deployment and configuration settings. The provider provides the networks, servers, storage, and other services. PaaS offerings facilitate the deployment of applications without the cost and complexity of buying and managing the underlying hardware and software and provisioning hosting capabilities [39].

PaaS offerings may also include facilities for application design, application development, testing, and deployment as well as services such as team collaboration, web service integration, and marshalling, database integration, security, scalability, storage, persistence, state management, application versioning, application instrumentation, and developer community facilitation.

MTC2 will use IaaS and PaaS layers that are supported in the United States by the Consolidated Afloat Networks and Enterprise Service (CANES) program. The fundamental goal of CANES is to bring IaaS and PaaS, within which current and future iterations of Tasking, Collection, Processing, Exploitation and Dissemination (TCPED) computing and storage capabilities will reside. CANES will provide complete infrastructure, inclusive of hardware, software, processing, storage and end user devices for Unclassified, Coalition, Secret and Sensitive Compartmented Information for all basic network services (email, web, chat, collaboration) to a wide variety of Navy surface combatants, submarines, Maritime Operations Centers, and Aircraft [25].

CANES is based around a cloud computing architecture that will host applications. Through this approach the Navy goal is to provide a common computing environment that will rely on commercial off the shelf equipment. By decoupling hardware from software, this will improve efficiency and reduce cost.

GCCS-M / MTC2 will be early adopter of the CANES implementation. Eventually other applications, systems and data sources will be added to provide an on-demand system synonymous

with cloud computing. By reducing complexity, eliminating proprietary systems and providing improved scalability and adaptability the the organization adopting MTC2 should realize more flexibility to changing conditions. Cost savings will be realized as the need for infrastructure, energy and staffing to manage the new system will be reduced [14].

The benefits of moving toward IaaS and PaaSe are, according to [29] :

1. Reduced complexity : existing capabilities have grown through independent acquisitions of components without an overall vision or architecture in mind. The emerging best practices for large-scale data center operations will drive simpler, more consistent infrastructures.
2. Better responsiveness : The ability to monitor the Global Information Grid (GIG) infrastructure across all applications, services, and user groups, along with the ability to respond dynamically to data storage and processing load will make it easier, faster and less expensive to allocate additional resources to meet new, unanticipated demands.
3. Shared Computer Infrastructure (CI) Resources : With the ability to share resources dynamically among applications, services, and user groups, peak transient CI demand for an application or service can be met by prioritization of the CI pool and by providing available infrastructure resources dynamically in response to priority uses.
4. Increased consolidation opportunities : With the ability to share processing and storage, the need to build excess capacity in every individual application's hardware in order to meet increased or unexpected user demand will be eliminated. This will have a dramatic positive effect on the overall cost of operations.

### 5.2.4 Software as a Service

SaaS is a software delivery model in which software and associated data are centrally hosted on the cloud. SaaS is typically accessed by users using a thin client via a web browser [40].

Several important changes to the software market and technology landscape have facilitated acceptance and growth of SaaS solutions [40]:

1. The growing use of web-based user interfaces by applications, along with the proliferation of associated practices continuously decreased the need for traditional client-server applications.
2. The standardization of web page technologies (HTML, JavaScript, CSS), the increasing popularity of web development as a practice, and the introduction and ubiquity of web application frameworks gradually reduced the cost of development.
3. The increasing penetration of broadband Internet access enabled remote centrally hosted applications to offer speed comparable to on-premises software.
4. The introduction and wide acceptance of lightweight integration protocols such as REST and SOAP enabled affordable integration between SaaS applications (residing in the cloud) with internal applications over wide area networks and with other SaaS applications.

SaaS eliminates the need to install and run the application on the customer's own computer.

Within the MTC2, web-services should offer the operator with tools to support the entire range of command and control functions [13] i.e planning, execution, monitoring and assessment.

## 5.3 Comparison of MTC2 and GCCS-M

The differences between COE and MTC2 are noticeable both at the architecture and functional point of view.

Concerning the data access and sharing, there is a paradigm shift to *need to share* in the MTC2, i.e, share your data first then process and share the information products. This is made so that some people have a quicker access to raw data to process them and generate more information products more quickly.

Also, where COE is a point to point data exchange between known producer and consumer, the MTC2 presents the data *as a Service* where one subscribes to a topic regardless of the data producer. Also, data are discoverable using a metadata registry, data catalogs, metadata discovery. MTC2 resolves many challenges of the COE architecture for data visibility, availability and understandability.

At the architectural level, there are several differences. At the infrastructure and platform levels, COE components and computer resources resides in each and every operator's work station. COE can be installed on a limited subset of available operation system and its components are tightly coupled with the OS on which it is installed. These systems lack scalability in computing power and storage.

MTC2 relies on the *as a Service* technology for both its infrastructure and platform. Using a cloud-like architecture, storage and computing power are scalable and can be shared and prioritized. This approach decouples hardware and software. This model also greatly reduces cost of software maintenance and deployment as everything is centralized.

At the software level, COE ensure software reuse and capability deployment through the use of Segments. Those segments must be built according to very strict specifications and must pass independent verification and validation to obtain a certain compliance level before being fielded. These new capabilities are then installed on the targeted computer to be available for a very limited number of operators. Each and every operator that wants or needs the new capability must have it installed on their own computer.

MTC2 relies on web services to deploy its capabilities. The mission specific applications reside on a remote computer, and are therefore available to anyone. The development of such web services obey well defined standards. These web services are discoverable, that means that searching within a service registry, one can find and use the capability it needs anywhere, anytime. This approach reduces software development cost, deployment time and the dependency of having to deal with a system administrator.

At the functionality level, the difference is also significant. GCC Systems provide simple tracks on a map, which supports situation awareness only. Also the information available about a particular track is limited to spatio-temporal information. Very little information is available about

tracks status, operational readiness and current tasking. Much more information will be readily available within MTC2. MTC2 will present the operator and decision makers with the so-called information halo. This information halo will provide dynamic Command and Control, anchored by enhanced situational awareness through rapid-access to relevant data associated to tracks (fuel status, ammunition loadout, manning, casual reporting, mission readiness, plans and tasking).

In addition to the enhanced situational awareness, the planned architecture and software deployment technology of MTC2 will address identified gaps in operational and tactical Maritime C2 requirements. As such the planned MTC2 will be [18] :

1. More complete by offering Planning, Execution, Monitoring and Assessment tools to allow commander's to seek answers to "What is," "What next," "What if," and "What was".
2. More collaborative across echelons and mission partners (joint and multi-national ) to allow Commanders to plan and re-plan, in near real time from decentralized locations that support the tactical situation.
3. More cohesive by offering Integrated, seamless and interoperable data exchange across ISR, IO, Cyber, combat systems, and combat support functions

This page is intentionally left blank.

# Part 6

# UAV Sensors and Recommendations

This section presents a review of UAV related documentation describing the roadmap plan for the development of future Unmanned Aerial System (UAS). It presents the current and future capabilities of the sensor payloads.

Based on those reviews, recommendations are provided on data to simulate in order to further test the CSD capabilities for the formation of maritime domain awareness.

## 6.1 UAV Sensor Payload

The role of the Maritime UAV system is to provide unmanned, long endurance, aerial reconnaissance, surveillance and target acquisition. In addition, the UAV can contribute to a comprehensive, real time, naval tactical picture for the ship's commander and naval head quarters.

A typical payload consists of a Maritime Patrol Radar (MPR) with multi-mode functions, an EO sensor with day/night capabilities and an optional Electronic Intelligence (ELINT) package. Often, maritime ISR UAV are also equipped with an AIS receiver. The payload package provides the necessary data for detection, classification and identification of surface vessels in the open sea [21].

The requirements for various payload capabilities can be grouped into five functional areas:

1. Imagery Intelligence (IMINT);
2. Signal Intelligence (SIGINT);
3. Measurement and Signatures Intelligence (MASINT);
4. communications;
5. and munitions.

Meteorological sensing stands outside this breakout, yet supports all of the others to some degree [30]. For the task under hand, which is related to maritime surveillance, we will focus on IMINT,

SIGINT. An overview of meteorological sensing capabilities will also be presented.

The following sections are built around information from the UAS Roadmap of the US army along with the website and documentation from the UAV manufacturers. The major UAV manufacturers considered are : Northrop Grumman, Rheinmetall, FLIR Systems, AeroVironment, BAE Systems, Elbit Systems, Denel Dynamics, L 3 WESCAM, Thales, Lockheed Martin, SAAB, General Atomics Aeronautical Systems, AAI Corporation and Israel Aerospace Industries.

### 6.1.1 Imagery Intelligence

IMINT is derived from photography, infrared sensors, synthetic aperture radar, and other forms of imaging technology. The following sections describe each type of available image type.

#### 6.1.1.1 Video/Electro-Optic/Infrared (EO/IR) Sensors

This sensor type is usually a camera which operates in the visible and/or infrared range. It provides full motion video, still imagery, image-intensified imagery, blended imagery and fused imagery. Other parts of this category are multispectral (tens of bands) and hyperspectral (hundreds of bands) imagery. They combine the attributes of panchromatic sensors to form a literal image of a target with the ability to extract more subtle information [33].

For unmanned systems, video data constitutes the only eyes-in-the-sky and thus the only link that a ground crew may have with what is actually going on in the vicinity of an unmanned air vehicle. It can be used for target detection and identification as well as providing information critical to situational awareness.

Still imagery produced by Charge-Coupled Device (CCD) daylight camera and infrared sensors (primary for night-based detection) also convey the same information as an HD video stream. However, still imagery cannot provide coverage for an area for several minutes and therefore might miss something in terms of situational awareness.

Hyperspectral Imagery (HSI) and Multispectral Imagery (MSI) systems are powerful tools in the field of remote sensing. While HSI systems collect at least 100 spectral bands of 10 - 20 nm width, MSI sensors are systems collecting less than 20, generally non-contiguous, spectral bands.

HSI systems have a very wide capability of spectral discrimination, while MSI systems are designed to support applications by providing bands that detect information in specific combinations of desirable regions of the spectrum. The number and position of bands in each system provide a unique combination of spectral information and are tailored to the requirements the sensor was designed to support [35].

A good understanding of background and object spectral signatures, and their dynamic behaviour in realistic environments is essential in the interpretation of HSI and MSI. Therefore the availability of good spectral libraries, i.e. collections of spectral reflectance and emissivity measured from materials of known composition, is of basic importance. Knowledge of the influence of the atmospheric conditions (temperature, pressure, humidity, haze or aerosol, wind speed and direction)

and of the solar radiation is extremely important in correctly interpreting the imagery [35].

Commercial satellite products (such as land remote-sensing satellite (LANDSAT) or Systeme Pour l'Observation de la Terre (SPOT)) have made MSI a mainstay of civil applications, with resolution on the order of meters or tens of meters. Systems designed for military applications are beginning to be fielded [31].

On the HSI side, military applications of HSI technology provide the promise for an ability to detect and identify particulates of chemical or biological agents. Passive HSI imaging of aerosol clouds could provide advance warning of an unconventional attack [31].

HSI and MSI systems both have the capability of producing images in which single pixels have spectral information content relevant to particulars of the scene under observation and have some common applications. Applications of interest for HSI and MSI sensors are [6]:

1. Target detection and identification: This includes targets such as military vehicles, camouflages and various man-made materials.
2. Land mapping applications: This includes the characterization of soil and vegetation.
3. Marine mapping applications: This includes beach characterization and near-shore bathymetry, as well as water color mapping.

For high or very high resolution systems, these kind of data can provide information related to situational awareness. Video from a UAV following a ship at sea can for instance give information about the presence of armed people and on the number of people on board.

However, since the number of bands acquired by a MSI system is less, the ground resolution of the image is larger than for its HSI counter-part where spectral resolution is higher.

Figure 6.1 and Figure 6.2 (from [31]) depict the technology forecast for still and video imagery.

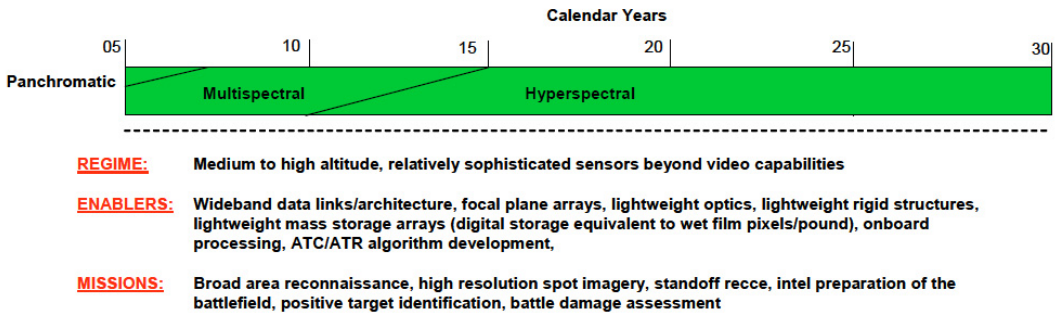


Figure 6.1: Still Imagery Technology Forecast.

In order to ensure interoperability, the standard format and associated metadata information of Electro-Optic/Infrared (EO/IR) sensors products are to be made compliant to the STANAG 4545 for secondary imagery and are stored under a NSIL_IMAGERY_VIEW within the CSD.

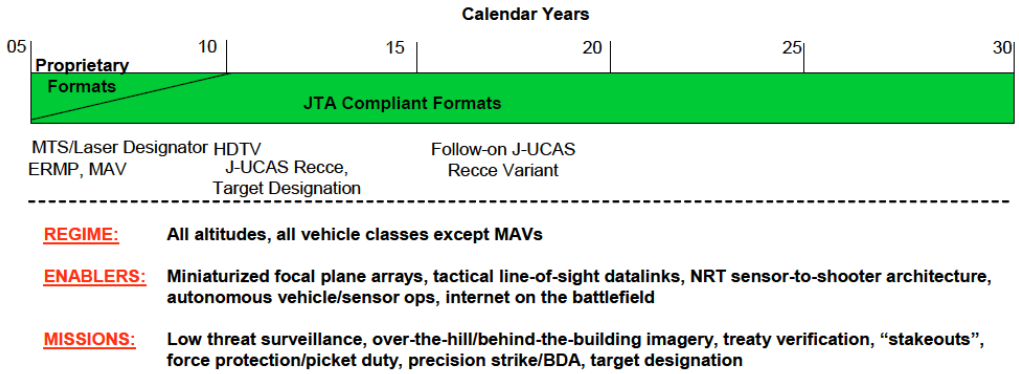


Figure 6.2: Videos Technology Forecast.

Video data, on the other hand, is compliant to the STANAG 4609 and stored under a NSIL_VIDEO_VIEW within the CSD.

Examples of currently fielded EO/IR sensor payloads are provided in A.1.

#### 6.1.1.2 Synthetic Aperture Radar (SAR)

Since airborne radars first appeared during World War II, they have been adapted to a wide variety of applications, from fire control and early warning to reconnaissance weather monitoring. Their key military value has been their ability to see farther than optical means and through conditions (night, clouds) which would otherwise deny their use. Conversely, their resolution is poorer, their use revealing to hostile forces, and their size, weight, and power a burden to their host aircraft, particularly to the smaller UAVs. Resolution has been significantly improved in the past two decades by the introduction of SAR, in which onboard processing uses the aircraft's forward motion to simulate a physically larger, fixed antenna, thereby increasing system gain and thus resolution.

Moving Target Indicator (MTI) is another mode of operation of a radar to discriminate a moving target against clutter. In contrast to another mode, stationary target indication, it takes advantage of the fact that the target moves with respect to stationary clutter.

Figure 6.3 (from [31]) depict the technology forecast for radar imagery.

SAR improvements are changing the nature of the product from simply an image or an MTI map to more detailed information on a target vehicle or battlefield. Current SAR systems can perform limited coherent change detection showing precise changes in a terrain scene between images. Use of phase data can improve resolution without requiring upgrades to the SAR transmitter or antenna, through data manipulation with advanced algorithms. These and other advanced SAR techniques require access to the full video phase history data stream and are often very processing intensive [31].

The information that can be extracted from SAR/PolSAR and GMTI imagery are :

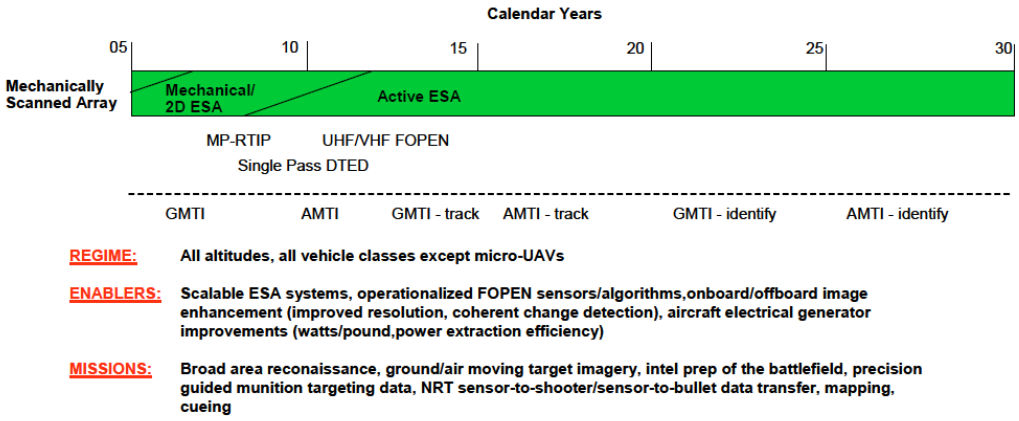


Figure 6.3: Radar Imagery Technology Forecast.

1. Target detection and identification: Stationary targets for SAR/PolSAR and moving target for GMTI. In addition, polarimetric SAR can highlight particular structures on the target that can help in target identification. Detection can be made at night or even under cloud cover.
2. Land mapping applications: This includes the characterization of soil and vegetation.
3. Marine mapping applications: Wind speed, ship wake.

In order to ensure interoperability, the standard format and associated metadata for which SAR / Polarimetric SAR (PolSAR) data must be compliant are defined in STANAG 4545 for secondary imagery format. These kinds of data will be stored under a NSIL_IMAGERY_VIEW within the CSD.

The standard to which the MTI data must comply are defined in the STANAG 4607. This kind of data will be stored under a NSIL_GMTI_VIEW within the CSD.

Examples of currently fielded SAR/GMTI sensor payloads are provided in A.2.

### 6.1.2 Signal Intelligence

Around the globe, modulated, encrypted and multiplexed signals are increasing in number, type and complexity, creating challenges for signal surveillance and exploitation. The SIGINT sensor capabilities provide detection identification, geo-location and copy of communications and non-communications emitters to provide situational awareness and intelligence on adversary. Due to security classification, SIGINT data will typically be processed at a secure facility.

A SIGINT system is expected to perform three functions: emitter mapping (geolocation of emitters), exploitation (signal content), and technical analysis of new signals. Moving closer to the emitter would allow lower-powered signals to be collected using readily available equipment, but also increases the threat to the collector aircraft an argument for UAV use [30] .

Figure 6.4 (from [31]) depicts the technology forecast for sensors related to signal intelligence.

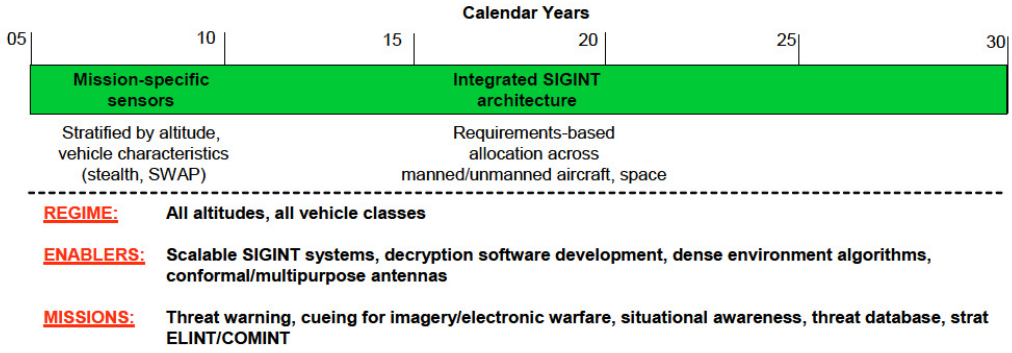


Figure 6.4: Signal Intelligence Technology Forecast

The information provided by ELINT and ESM data are generally in the form of track data, where the emitter and active sensor suite have been mapped to a particular kind of platform using *a priori* database. This kind of information is usually transmitted and stored in the CSD in the form of Tactical Data Link. It complies to the standard defined in the STANAG 5516 and is stored in the CSD under a NSIL_TDL_VIEW

Examples of currently fielded SIGINT sensor payloads are provided in A.3.

### 6.1.3 Automatic Identification System Receiver

The AIS is a broadcast system used on ships and by vessel traffic services for identifying and locating vessels by electronically exchanging data with other nearby ships, AIS base stations, and satellites. When satellites are used to detect AIS broadcasts then the term Satellite-AIS is used.

AIS information supplements maritime patrol radar in UAV payload. It is normally used to perform monitoring of the maritime zone and also to cue more advanced EO/IR sensors.

The information content in the AIS messages is very rich. It includes but is not limited to :

1. The vessel's Maritime Mobile Service Identity;
2. Navigation status;
3. Rate of turn;
4. Speed;
5. Position;
6. Course;
7. True heading;
8. True bearing;

9. Timestamp of the message;
10. IMO ship identification number;
11. Radio call sign;
12. Name of the ship;
13. Type of ship/cargo;
14. Dimensions of ship;
15. Type of positioning system;
16. Draught of ship;
17. Destination;
18. ETA (estimated time of arrival) at destination;

There is currently no STANAG for AIS data. Some XML formats (NIEM, exactEarth, etc.) exist but none has been accepted as the baseline standard for interoperability for a joint operation. Note that AIS meets the National Marine Electronics Association (NMEA) standard.

Examples of currently fielded SIGINT sensor payloads are provided in A.4.

### 6.1.4 Maritime Patrol Radar

All Medium Altitude - Long Endurance (MALE) UAV are usually equipped with such type of MPR [33]. The information provided by a MPR is generally tracks with internal track number. It does not give any identity information. Examples of currently fielded SIGINT sensor payloads are provided in A.5.

### 6.1.5 Light Detection and ranging

Light Detection And Ranging (LIDAR) sensors provide high-resolution, 3-dimensional (3D) geospatial data. LIDAR data can be used as a stand-alone product, or as an accurate foundation for rectifying and draping high-resolution imagery.

According to [28], LIDAR provides a way to see urban areas in rich 3-D views that give tactical forces unprecedented awareness in urban environments. LIDAR data is both high-resolution and high-accuracy, enabling improved battlefield visualization, mission planning and force protection. LIDAR also supports automated extraction of urban features like buildings and trees-a critical technological improvement for constructing simulation databases rapidly. Typical LIDAR data sets are 1-meter resolution, although higher resolutions have been collected where possible. The standard projection is Universal Transverse Mercator, and the standard format for gridded data is GeoTIFF.

There are normally 5 files for every LIDAR data set [28]:

1. First Return Digital Elevation Model (DEM) - 32-bit floating point gridded matrix

2. Last Return DEM - 32-bit floating point gridded matrix
3. Intensity Image (int) - 8-bit
4. Bare Earth DEM - 32-bit floating point gridded matrix
5. Merged Intensity-Color Coded Shaded Relief Image (mrg) - 24-bit

In addition, LIDAR maybe used for explosive hazard detection and weather predicting (e.g Doppler LIDAR could provide data such as cloud density and wind speed as well as vertical wind profile). Also, some new multispectral LIDAR are designed to detect and image effluents that are associated with chemical and biological warfare agents [33].

Even if there is no standard available for this kind of data, we could expect to be able to store them in the format compliant to the STANAG 4545 for secondary imagery or very close and are stored under a NSIL_IMAGERY_VIEW within the CSD.

### 6.1.6 Laser Radar

The Laser Radar (LADAR) performs three dimensional imaging. It as the capability to look through cover such as trees, foliage and camouflage. LADAR will produce a virtual picture to reliably identify previously hidden targets (tanks or other vehicles). This technology also has the potential for assisting with explosive hazard detection.

Even if there is no standard available for this kind of data, we could expect to be able to store them in the format compliant to the STANAG 4545 for secondary imagery or very close and are stored under a NSIL_IMAGERY_VIEW within the CSD.

### 6.1.7 Sensors in the context of maritime surveillance operations

In the context of maritime surveillance, some sensors are of particular importance. In addition to EO/IR sensors, that are the eye of the UAV for the ground crew and is not optional, the following sensors should be prioritized when building a UAV for maritime surveillance :

- **AIS receiver** : All ship above a certain gross tonnage are equipped with an AIS which make this sensor a valuable asset in a UAV payload.
- **Maritime patrol radar** : Can be used to detect smaller ship not equipped with AIS, in all-weather condition, or validate an AIS detection.
- **SAR imagery** : Can provide all-weather target detection and provide some information on the detected of object.
- **SIGINT Sensor suite** : This sensor can provide identity information based on the measurement of electronic signatures. Mainly used for military application.

AIS and MPR are usually used to cue the UAV towards a detected vessel so that the EO/IR sensor can take an image or a video feed of it to provide enhanced situation awareness.

Currently, the U.S. Navys Broad Area Maritime Surveillance (BAMS) sensor suite, including Northrop Grummans Multi-Function Active Sensor radar, Raytheons MTS-B with an AIS for ship tracking, and an Electronic Support Measures (ESM) system is the largest UAV program dedicated to maritime surveillance [24]. This sensor suite developed as part of it is composed of all the sensor types identified above which will make the UAV equipped with it the perfect candidate for maritime surveillance mission.

### 6.1.8 Sensors in the context of maritime environmental operations

Some other payload can be included in the UAV to gather information about the state of the ocean and the environmental conditions. Many of these sensors produce point-based measurements. Coupled with a GPS, these point-based measures can be georeferenced and store for further processing at the ground station using dedicated software. The following list presents the type of environmental parameters that can be gathered using different sensors as well as their intermediate and end products.

- **Wind Speed/Direction** : These parameters are measured using a pitot, which when coupled with a GPS measurement provides point-based measurements of wind speed and direction at constant intervals. These points are then usually processed at ground station and included in the form of directional vectors as overlay on an image product (see figure 6.5, from `http://marine.rutgers.edu/~shann16/page1.html`).
- **Relative Humidity, Temperature and Pressure** : These parameters are measured using digital meteorological sensors (thermometer, hygrometer and barometer), which when coupled with a GPS measurement provides point-based measurements of wind speed and direction at constant intervals. These points are usually further processed using dedicated and/or specialized software. The end products are usually presented as map overlays and can therefore be included in the NSIF format.
- **Ocean /Ice Skin Temperature** : Measured by an infrared thermometer or by an Infrared sensor, the end product is usually a heat map in an image format.
- **Ocean/Ice Images** : These are standard imagery products taken by a digital camera.
- **Shortwave Radiation** : Measure by an Up and Down-Looking Pyranometer, which when coupled with a GPS measurement provides point-based measurements, These points are usually further processed using dedicated and/or specialized software. It produces an image or heat map that display the shortwave radiation intensity.
- **Longwave Radiation** : Measure by an Up and Down-Looking Pyranometer, which when coupled with a GPS measurement provides point-based measurements, These points are usually further processed using dedicated and/or specialized software. It produces an image or heat map that display the longwave radiation intensity.
- **Altitude and Surface Waves** : These can be measured by a laser altimeter. This kind of sensor provides georeferenced point-based measurement of the altitude of the sensor relative to the surface from which the altitude of the surface or the surface waves of the sea are

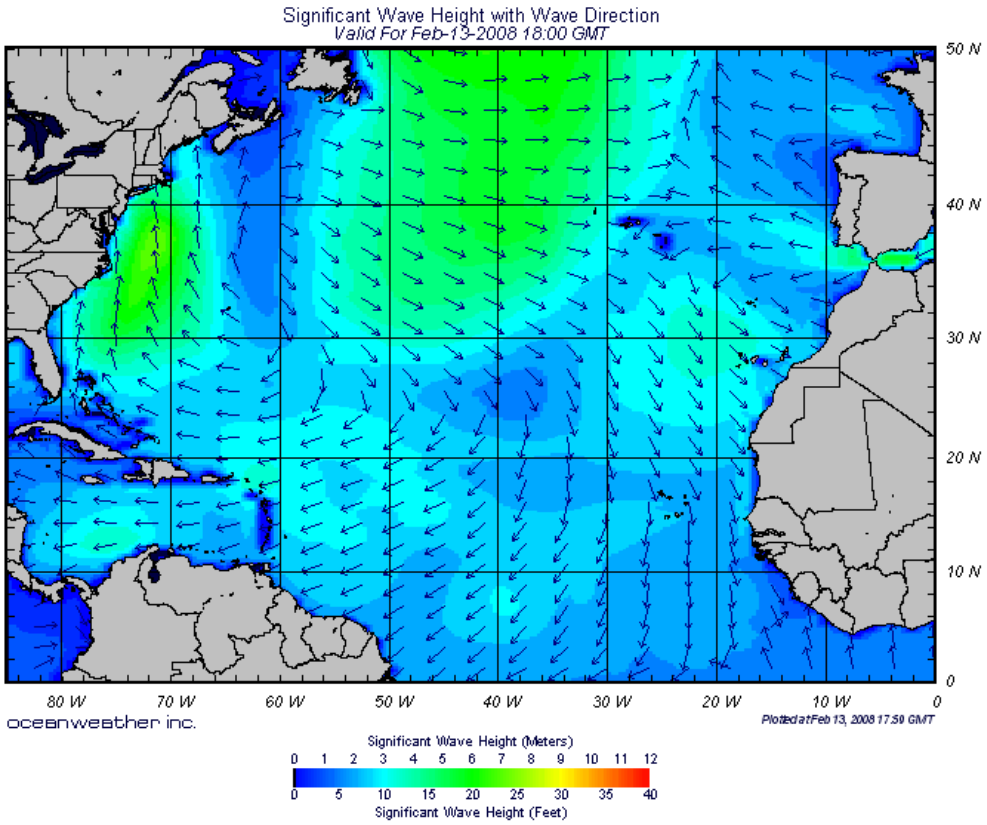


Figure 6.5: Ocean weather image presenting waves heights and wind speed and direction

then derived. These points are usually further processed using dedicated and/or specialized software. It produces a digital elevation model, in the case of altitude, or an image or heat map that display the waves heights (see figure 6.5).

All these parameters can contribute to the Recognised Environmental Picture which is defined as *a complete and seamless depiction of geospatial, oceanographic and meteorological information designated for the planning and conduct of joint operations in a specific area at a specific time and which supports the unity of effort across the joint Battlespace* [12].

Meteorological data are not defined as a standard product in the CSD and should not be in the near future. As stated in [12], NATO is currently seeking to develop NATO Environmental Functional Services. Currently, for distribution of the geospatial information NATO has adopted the service model described by the Open Geospatial Consortium, including Web Map Service, Web Feature Service and Web Coverage Service, as well as Web Map Tile Service. It is expected that in case of oceanographic and meteorological information the standard services need to be extended with dimension parameters (e.g. time, elevation). For non-geospatial information (e.g. texts, diagrams, multi-media files) general web services, e.g. based on the XML, are to be used [12].

### 6.1.9 Sensor and information products in relation to the CSD capabilities

It is mentioned in [33] that all UAV data products will be STANAG compliant in order to assure interoperability between coalition members. As such, all GMTI products are compliant with the STANAG 4607. Compliance with the STANAG 4609 provides interoperability with the FMV and large volume of streaming data. Also, for stored information in the NATO CSD, the STANAG 4559 defines the standard interfaces to allow the sharing of intelligence information. The UAV data products shall be compliant to standards defined in the STANAG that relates to them.

### 6.1.10 Summary and Recommendations for data simulation

This section summarizes the current and/or near term forecasted capabilities of the sensor types. It also provides recommendations about the data to be simulated.Survey of payload currently in development is provided in A.6, while a survey of currently deployed UAVs and their possible payload configuration is provided in A.7.

Table 6.1 summarizes the current and/or near term forecasted capabilities of sensor payloads.

Table 6.1: Current and near-term capabilities of sensor types

| Sensor | Current and/or near-term capabilities |
|---|---|
| Radar | • SAR / PolSAR- Target detection and recognition, environmental conditions, terrain mapping;<br>• GMTI - Track and Identity;<br>• AMTI - Track;<br>• UHF/VHF FOPEN - Target detection. |
| EO/IR/MSI | • Target detection and identification;<br>• Terrain mapping;<br>• Marine mapping. |
| AIS | • Ship self report information, rich identity and positional content. |
| MPR | • Detection position. |
| SIGINT | • Target detection and recognition. |
| LADAR | • Digital elevation model;<br>• Target detection and recognition. |

| | |
|---|---|
| HSI | • Target detection and identification;<br>• Terrain mapping;<br>• Marine mapping. |
| Video | • Enhanced situational awareness (example : visual of crew on deck)<br>• Integrated target designation<br>• Autonomous operation<br>• Target search/ID functions |

With regards to recommendation on data to simulate we can state the following. First, the currently available data is video and metadata, suggesting that still EO/IR imagery should already be available. Also, in the context of maritime ISR, LADAR and HSI are not really useful in the context of maritime ISR.

Therefore, considering the sensor and information usefulness in maritime surveillance, probability of being part of payload and how relatively easy it is to simulate, ordered in priorities, one should simulate the following :

- AIS receivers;
- Maritime patrol radar;
- EO/IR and SAR / Inverse SAR;
- SIGINT components : ESM principally and IFF to a lesser extent.

Imagery might be difficult to simulate but, based on past experience, image databases already exist and should be available within the DRDCs (e.g Foward Looking Infrared and SAR/Inverse SAR).

# Part 7

# Difficulties and Conclusion

## 7.1 Difficulties encountered

Several difficulties were encounter during this Call-up. Most of them were related to incomplete documentation.

- The installation of the software furnished as GFI, the operations were not always obvious, the documentation was not always accurate and several work arounds had to be found.
- SC2PS does not install correctly under Windows 7 and no error window is ever thrown to provide the user with a reason of non-working functionalities.
- The CSD and the API documentation was not complete or inexistant which reflected in the time it took to realize some of the tasks.
- The MTC2 documentation was not available as GFI during the time frame of this call up. As such, it was necessary to devote a certain amount of time to finding and digesting several technical documents and papers, presentations and news articles scattered in the world wide web.
- The CSD API version 2.0 provided in the lib directory of the CSD is not working properly for product creation. No error is thrown at the developer. The CSD API version 2.3, found in the lib directory of SC2PS solved the problem without code modification.

## 7.2 Conclusion

This section concludes the final report for Call-up 11 and Call-up 3. Within these Call-ups, several aspects of the CSD and UAV were investigated in the context of maritime situation awareness.

- A test bed has been developed to send UAV data into a CSD instance and synchronize it

with another CSD instance. A slowdown mechanism was putted in place to mimic network disturbance by limiting the bandwidth available between CSD instances.

- Investigation has been made to assess the possibility of storing motion imagery data provided by a UAV in the NPR (GPW v2.0). It was determined that the two systems are sufficiently different and that unless major modifications are made to the NPR or post-processing are made on the UAV data, these are not compatible at least for video imagery.
- AIS data in relation with the CSD was also investigated. AIS decoders were implemented and the decoded messages were successfully mapped in the NIEM XML data exchange standard. However, some work around had to made to store the data in the CSD as no standard are actually adopted for AIS metadata/data storage and interoperability.
- A comparison of the current GCCS-M with the upcoming MTC2 highlighted their differences at the architectural and functional levels.
- A survey of current and future UAV sensor payloads was made, which led to recommandations on which data would be best to simulate in the context of maritime surveillance.

# Bibliography


[1] NATO Standardization Agency. Stanag 5516 tactical data exchange - link 16. Technical report, NATO, 2006.

[2] NATO Standardization Agency. Nato ground moving target indicator format (gmtif) stanag 4607 implementation guide. Technical report, NATO, 2008.

[3] NATO Standardization Agency. Nato motion imagery (mi) stanag 4609 (edition 3) implementation guide. Technical report, NATO, 2009.

[4] NATO Standardization Agency. Nato standard isr library interface (nsili) edition 3. Technical report, NATO, 2010.

[5] NATO Standardization Agency. Nato secondary imagery format (nsif) stanag 4545 (edition 2) implementation guide. Technical report, NATO, 2013.

[6] Jean-Pierre Ardouin, Josee Levesque, Vincent Roy, Yves Van Chestein, and Anthony Faust. Demonstration of hyperspectral image exploitation for military applications. In *Remote Sensing - Applications*, pages 493–516. 2012.

[7] Jaroslav Blaha. Information technology standards and standardization. chapter A Standards-based Common Operational Environment, pages 152–167. IGI Global, Hershey, PA, USA, 2000.

[8] Deidre Briggs. Nces: Enabling the dod net-centric data and service strategies. `1105govinfoevents.com/events/2009/sds09/colbriggspresentation.pdf`, 2009. [Online; accessed November-2013].

[9] Chris Bursk. Independent research of the united states defense information infrastructure common operating environment. `http://www.reocities.com/cbursk/COEresearch.html`, 2003. [Online; accessed January-2014].

[10] Ada Information Clearinghouse. Plug and play for the warfighter. `http://archive.adaic.com/news/Newsletter/1997/fall/7.htm`, 1997. [Online; accessed January-2014].

[11] U.S. Army Information Systems Engineering Command. Automated information systems (ais) design guidance long-haul transmission systems. `http://www.fas.org/spp/military/docops/army/lhaul/Lhfinweb.htm`, 1998. [Online; accessed November-2013].

[12] NATO Communication and Information Agency. Market survey co-13524-ms - functional services for environmental support to operations, July 2012.

[13] Keith Debban. Maritime tactical command and control (mtc2) industry day. http://www.defenseinnovationmarketplace.mil/resources/USN%202011%2011%209%20MTC2%20Industry%20Day.pdf, 2011. [Online; accessed November-2013].

[14] DevTome. Introduction of cloud computing to the united states navy. http://devtome.com/doku.php?id=navy_cloud_computing, 2013. [Online; accessed November-2013].

[15] Gregory Frazier. The dii coe: An enterprise framework. *Cross Talk*, October 2001.

[16] Flight Global. Watchkeeper uav edges towards uk acceptance. http://www.flightglobal.com/news/articles/watchkeeper-uav-edges-towards-uk-acceptance-391499/, 2013. [Online; accessed October-2013].

[17] Northrop Grumman. An/zpy-2 multi-platform radar technology insertion program (mp-rtip). http://www.northropgrumman.com/Capabilities/MPRTIP/Pages/default.aspx, 2013. [Online; accessed October-2013].

[18] Donald Harder. Moving navy command and control into the future. http://www.public.navy.mil/spawar/Press/Documents/Publications/1.30.2013_AFCEA[150_PM].pdf, 2013. [Online; accessed November-2013].

[19] HomelandSecurity-Technology.com. Sagem completes flight test of patroller drone system. http://www.homelandsecurity-technology.com/news/newssagem-completes-flight-test-of-patroller-drone-system, 2012. [Online; accessed October-2013].

[20] Jerome Hudson. Software agents and the defense information infrastructure: reengineering the acquisition process. Technical report, DTIC Document, 1998.

[21] Israel Aerospaces Industries. Iai/malat solutions for the maritime arena. http://www.iai.co.il/sip_storage/FILES/3/38133.pdf, 2013. [Online; accessed October-2013].

[22] Anthony W. Isenor and Eric Dorion. The use of gccs in the canadian navy and its relationship to c2iedm. Technical Report DRDC-ATLANTIC-TM-2004-197, DRDC-Atlantic, Dartmouth, February 2005.

[23] Janes. Ausa 2013: Ultraeagle elint system makes its debut. http://www.janes.com/article/28963/ausa-2013-ultraeagle-elint-system-makes-its-debut, 2013. [Online; accessed October-2013].

[24] Earth Imaging Journal. Uav market set for 10 years. http://eijournal.com/uncategorized/uav-market-set-for-10-years, 2011. [Online; accessed October-2013].

[25] US Navy. Exhibit r-2, rdt and e budget item justification: Pb 2013 navy. http://www.dtic.mil/descriptivesum/Y2013/Navy/stamped/0303238N_7_PB_2013.pdf, 2012. [Online; accessed November-2013].

[26] U.S. Navy. Multimode sensor / seeker. http://www.onr.navy.mil/en/Media-Center/Fact-Sheets/Multi-Mode-Sensor-Seeker.aspx, 2013. [Online; accessed October-2013].

[27] Depertment of Defence. Nces tech guide. https://metadata.ces.mil/dse/ns/ces/techguide/printable_techguide.doc, 2008. [Online; accessed January-2014].

[28] US Army Corps of Engineers. Lidar data. http://www.agc.army.mil/Media/FactSheets/FactSheetArticleView/tabid/11913/Article/10229/lidar-data.aspx, 2013. [Online; accessed December-2013].

[29] Office of the Chief Information Officer. Department of defense information enterprise architecture version 1.1. Technical report, Department of Defence, 2009.

[30] Office of the Secretary of Defence. Unmanned aerial vehicles roadmap 2000 - 2025. Technical report, Department of Defence, 2001.

[31] Office of the Secretary of Defence. Unmanned aircraft systems roadmap 2005 - 2030. Technical report, Department of Defence, 2005.

[32] Margaret Rouse. Infrastructure as a service. http://searchcloudcomputing.techtarget.com/definition/Infrastructure-as-a-Service-IaaS, 2013. [Online; accessed December-2013].

[33] U.S Army UAS CoE Staff. U.s army roadmap for unmanned aircraft systems 2010 - 2035. Technical report, U.S Army UAS Center of Excellence, 2010.

[34] OODA Technologies. Maritime situational awareness research infrastructure (msari): Requirements and high level design. Technical report, Defence Research and Development - Atlantic, March 2012.

[35] Fabrizio Vagni. Survey of hyperspectral and multispectral imaging technologies. Technical report, NATO RTO, 2007.

[36] Wikipedia. Data as a service. http://en.wikipedia.org/wiki/Data_as_a_service, 2013. [Online; accessed December-2013].

[37] Wikipedia. Mil-std-6011. http://en.wikipedia.org/wiki/TADIL-A, 2013. [Online; accessed January-2014].

[38] Wikipedia. National information exchange model. http://en.wikipedia.org/wiki/National_Information_Exchange_Model, 2013. [Online; accessed December-2013].

[39] Wikipedia. Platform as a service. http://en.wikipedia.org/wiki/Platform_as_a_service, 2013. [Online; accessed December-2013].

[40] Wikipedia. Software as a service. http://en.wikipedia.org/wiki/Software_as_a_service, 2013. [Online; accessed December-2013].

[41] Wikipedia. Tadil-j. http://en.wikipedia.org/wiki/TADIL-J, 2013. [Online; accessed January-2014].

[42] Wikipedia. Nasa world wind. http://en.wikipedia.org/wiki/NASA_World_Wind, 2014. [Online; accessed March-2014].

[43] PR News Wire. Raytheon's seavue xmc showcases innovative maritime radar solutions. http://www.prnewswire.com/news-releases/raytheons-seavue-xmc-showcases-innovative-maritime-radar-solutions-222503311.html, 2013. [Online; accessed October-2013].

This page is intentionally left blank.

# Appendix A

# Summary of UAV current sensor payload in the US Navy

## A.1 Example of currently fielded EO/IR payloads

As current UAV payload, the AF Predator and Army Hunter use real-time video systems mounted in turrets. While initial systems were derivatives of commercial products, retrofit with sensors and designators specific to military applications is underway. The Air Force is integrating the Multispectral Targeting System-A EO/IR laser target designators/illuminators into Predator; in the same vein, the Army is planning to integrate a designator into Shadow (RQ-7B).

There is also the Global Hawk Integrated Sensor Suite (ISS). The ISS consists of a SAR imaging radar with GMTI mode and an EO/IR sensor that produces still imagery.

In the near future, the Advanced EO/IR Unmanned Aircraft (UA) sensor is a high resolution, highly stabilized EO/IR sensor being developed for Army UA by the Army's Night Vision Electronic Sensors Directorate. It consists of a multi field-of-view sensor that will provide greater standoff ranges and highly stabilized gimbals that allow for an increase in the area of coverage. Its all digital output is Joint Technical Architecture compliant.

## A.2 Example of currently fielded SAR payloads

The following list presents a subset of currently fielded SAR payloads onboard of UAS [31]:

- Global Hawk ISS Radar. Dedicated Global Hawk, SAR capable of spot, search, and GMTI modes; 1-foot resolution.
- LYNX. A tactical radar, deployed in various configurations on both manned and unmanned aircraft, most recently on the Army's I-GNATs. LYNX has a resolution of 4 inches in the spotlight mode, and provides GMTI and coherent change detection capabilities.

- TESAR. Tactical Endurance SAR (TESAR) is a strip mapping SAR providing continuous 1 foot resolution imagery. TESAR is flown on Predator.
- Tactical UAV radar (TUAVR). A 63-pound SAR/MTI radar for use on Army UA. Provides 1 foot resolution imagery in strip and spotlight modes and an integrated GMTI capability. The radar has been demonstrated on Hunter UA.
- MISAR. Developed by EADS, this small, Ka-band radar weighs approximately 10 pounds. It has been demonstrated on the German LUNA UA as well as on U.S. helicopters.

The Multi-Platform Radar Technology Insertion Program (MP-RTIP) should result in a more capable SAR active electronically steered antenna within this decade. Larger UA, such as Global Hawk for the Air Force and potentially for the Navy's BAMS role, are one intended recipient of this technology [31]. The MP-RTIP radar uses active electronically scanned array (AESA) technology and commercial off-the-shelf hardware to deliver long range, very high-resolution SAR, GMTI capabilities and air target tracking. A Northrop Grumman - Raytheon team is developing the radar for the United States Air Force Electronic Systems Center. MP-RTIP, is currently in the system development and demonstration phase [17].

## A.3 Example of currently fielded SIGINT payloads

An example of SIGINT payload is the Ultra Electronics Tactical Communications Systems Ultra-EAGLE ALR-510 airborne tactical ELINT system. Designed particularly for UAV and smaller manned airborne platforms, the system provides wideband microwave search and intercept with a phased interferometry system providing a direction finding accuracy [23].

## A.4 Example of currently fielded AIS payloads

For instance, Sagem's long-endurance Patroller drone system is equipped with an AIS receiver, enabling the real-time monitoring of maritime traffic over a large zone [19].

The Triton MQ-4C, of the the Navy's BAMS program is also equipped with an AIS receiver. Some of the Navy's Small Tactical UAS, such as the RQ-21A Interceptor, are also fitted with an AIS receiver.

## A.5 Example of currently fielded MPR payloads

The Multi-Function Active Sensor is an active electronically scanned array radar that provides a 360-degree field of view and is part of the current payload of the MQ-4C Triton has an MPR functionality. Also the MB-5B Hunter or the MQ-1C Gray Eagle (upgrade of the General Atomics Predator) ) are equipped with an MPR.

## A.6 Sensor payload in development

This section present some system that are currently in development. Essentially, these system incorporate improvements or increase in performance of current sensor technologies.

For instance, Sagem (http://www.sagem.com) said the Patroller drone was equipped with a new version of the Sagem Euroflir 350 gyrostabilized optronic pod, including an HDTV channel, third-generation HD infrared channel and a laser rangefinder; an AIS receiver, enabling the real-time monitoring of maritime traffic over a large zone; and a beacon for detecting distress signals [19].

The Multimode Sensor/Seeker (MMSS) [26] will deliver a sensor suite for use by future reconnaissance and targeting platforms that integrates eye-safe LADAR, visible and mid-wave infrared imaging sensors with a laser rangefinder / designator. The project will implement Automatic Target Recognition algorithms to perform target detection, classification, and identification of maritime targets and shore facilities. MMSS is also investigating concepts for its incorporation into the Multi-Spectral Targeting System-A system.

Raytheon's SeaVue XMC (eXpanded Mission Capability) radar is a next-generation maritime situational awareness package developed during for the Ocean Surveillance Initiative program sponsored by the US Navy. SeaVue XMC provides both radar and mission system capabilities, including automatically detecting, tracking, and sorting thousands of maritime targets simultaneously and correlating those radar tracks with AIS contacts. The system geographically registers radar detections to AIS data and digital nautical chart features for more precise tracking, threat location, and accurate cross-sensor cueing to the electro-optic system. It is currently operational on the United States (US) Customs and Border Patrol Dash 8 and the P-3 aircraft [43].

Increased onboard processing will also be a technology for future UAVs. Autonomous operation is a current capability-push by the Navy in the Office of Naval Research's autonomous operation Future Naval Capability initiative and by the Air Force as part of the Air Force Research Laboratory's Sensorcraft initiative.

## A.7 Summary of UAV ant their current payloads

Table A.1 provides an overview of currently fielded UAVs and their possible payload configuration.

Table A.1: Currently deployed UAV and available sensor payload

| UAV | Sensor payload |
|---|---|
| Predator-A (MQ-1) | • Raytheon Multispectral Targeting System (MTS) AN/AAS-52<br>• Northrop Grumman TESAR Synthetic Aperture Radar |

| | |
|---|---|
| Predator-B (MQ-9 Reaper) | • Two-Colour DLTV Television<br>• High Resolution FLIR<br>• Synthetic Aperture Radar |
| Triton (MQ-4C) | • 360-degree field of regard (FOR) sensors<br>• EO/IR video and still imagery<br>• MTS-B multispectral targeting system<br>• Multifunction active sensor (MFAS) electronically steered array radar (SAR / MPR)<br>• AIS receiver<br>• AN/ZLQ-1 ESM system |
| Global Hawk | • Synthetic Aperture Radar (SAR)<br>• Moving Target Indicator (MTI)<br>• Electro-optical, NIIRS 5.5 / 6.5 (WAS/Spot)<br>• Infrared, NIIRS 5.0 / 6.0 (WAS/Spot)<br>• Target Coverage, 1900 Spot target per day |
| Heron TP | • EO/IR/LRF<br>• SAR<br>• Maritime Patrol Radar<br>• ELINT<br>• COMINT<br>• AIS receiver |
| ScanEagle | • EO/IR Camera<br>• SAR |
| RQ-11 Raven | • Dual-forward and side-look EO camera nose<br>• Forward and side-look camera-nose<br>• Thermal IR |

| | |
|---|---|
| Heron 1 | • Electro-optical (TV and IR Combi or triple sensors TV/IR/LD)<br>• Synthetic Aperture Radar (SAR)<br>• Maritime Patrol Radar (MPR)<br>• COMINT and ESM capability<br>• Communication Relay package<br>• Integrated ATC package<br>• AIS receiver |
| Kahu 2-eb | • Can carry a range of sensor packages up to 3 kg |
| Hawk | • 8 mega pixel still camera or full motion video 2x digital zoom |
| CamCopter S-100 | • Stabilized Daylight/Infrared Gimbal<br>• Synthetic Aperture Radar<br>• Light Detection and Ranging<br>• Multispectral Imaging<br>• Ground Penetrating Radar |
| Rheinmetall KZO | • FLIR<br>• SAR<br>• ESM |
| Hermes 900 Maritime Patrol | • Gabianno T-200 MPR and SAR/GMTI<br>• AIS Receiver<br>• DCoMPASS EO/IR/Laser<br>• AES 210 V - ESM/ELINT with IFF Transponder<br>• Skyfix / Skyjam - COMINT/DF and optional COM-JAM system |
| Seeker 400 (Denel Dynamics) | • Zeiss LEO III (electro-optic (EO), infra-red (IR))<br>• Electronic Surveillance Payload (ESP)<br>• ELINT<br>• SAR<br>• MPR |

An other UAS in service is the Watchkeeper. It is an UK UAV, carrying an electro-optical/infrared sensor and synthetic aperture radar/ground moving target indication payload, the Watchkeeper air vehicle is an evolution of the Hermes 450 airframe [16].

One of the latest UAS released by Lockheed Martin Corporation is the stealth Q-170 Sentinel. An electro-optic and infra-red sensors are incorporated in the upper surface of the RQ-170 wings. The RQ-170 Sentinel is fitted with an active electronically scanned array radar, synthetic aperture radar and signal intelligence in its belly fairings.