Experiment scenarios, prototypes and report - Iteration 2

Emil Andriescu, Amel Bennaceur, Antonia Bertolino, Paul Grace, Trân Huynh, Marta Kwiatkowska, Bengt Jonsson, Antoine Léger, Animesh Pathak, Pierre-Guillaume Raverdy, et al.

CONNECT

Emergent Connectors for

Eternal Software Intensive Networked Systems

ICT FET IP Project

Deliverable D6.3

Experiment scenarios, prototypes and report – Iteration 2

SEVENTH FRAMEWORK PROGRAMME

http://www.connect-forever.eu

Inria · docomo DOCOMO Euro-Labs · Amalencic · LANCASTER UNIVERSITY · THALES · Universita' dell'Aquila · technische universität dortmund · University of Oxford · Uppsala Universitet

Project Number	:	231167
Project Title	:	CONNECT – Emergent Connectors for Eternal Software Intensive Networked Systems
Deliverable Type	:	Report

Deliverable Number	:	D6.3
Title of Deliverable	:	Experiment scenarios, prototypes and report – Iteration 2
Nature of Deliverable	:	R/P
Dissemination level	:	PU
Internal Document Number	:	V3
Contractual Delivery Date	:	M36
Actual Delivery Date	:	24 February 2012
Contributing WPs	:	WP6
Editor(s)	:	Trân Huynh (THA), Antoine Léger (THA)
Author(s)	:	Emil Andriescu (Ambientic-Inria), Amel Bennaceur (Inria), Antonia Bertolino (CNR), Paul Grace (LANCS) Trân Huynh (THA), Marta Kwiatkowska (UOXF), Bengt Jonsson (UU), Antoine Léger (THA), Animesh Pathak (Inria), Pierre-Guillaume Raverdy (Ambientic), Rachid Saadi (Ambientic), Roberto Speicys-Cardoso (Ambientic), Daniel Sykes (Inria), Massimo Tivoli (UNIVAQ)
Reviewer(s)	:	Valérie Issarny (Inria)

Abstract

The task of WP6 is to evaluate the CONNECT technologies under realistic situations. To achieve this goal, WP6 concentrated its 3rd year effort on the development of a main scenario in the context of GMES, which requires the connection of highly heterogeneous and independently built systems provided by the industry partners.

The resulting scenario allows the consortium to assess the validity of CONNECT claims and to investigate the exploitation of CONNECT technologies in the context of the integration of real systems.

Another objective of this report is to provide a first assessment of CONNECT solutions against the project's objectives stated in the DoW. The proposed assessment spans: (i) the project's overall objective of enabling on-the-fly interoperability among heterogeneous networked systems as well as (ii) the project's specific objectives related to the foundations and associated enablers to be elaborated for learning and reasoning about the interaction behaviours of networked systems and for synthesizing mediators so as to make systems interoperate.

Keyword list

Assessment, CONNECTor, CONNECTability, Experiment, GMES, CONNECT architecture, CONNECTor algebra, Mediator synthesis, Protocol learning.

Document History

Version	Type of change	Author(s)
0.1	Initial version	Trân Huynh (THA), Antoine Léger (THA)
0.2	add affordances and bpel	Antoine Léger (THA)
0.3	Update interfaces, affordances, sawsdl & bpel	Antoine Léger (THA)
0.4	Add Inria contributions	Antoine Léger (THA), Animesh Pathak (Inria), Amel Bennaceur (Inria), Daniel Sykes (Inria)
0.5	Add Ambientic contributions	Antoine Léger (THA), Emil Andriescu (Ambientic-Inria), Pierre-Guillaume Raverdy (Ambientic), Roberto Speicys-Cardoso (Ambientic), Rachid Saadi (Ambientic)
0.6	Use new doc template	Antoine Léger (THA)
0.7	Elaborate assessment part	Daniel Sykes (Inria), Paul Grace (LANCS), Marta Kwiatkowska (UOXF), Massimo Tivoli (UNIVAQ), Bengt Jonsson (UU), Antonia Bertolino (CNR), Antoine Léger (THA)
0.8	Insert assessment part	Antoine Léger (THA), Daniel Sykes (Inria)
0.9	Update BPEL schemas	Antoine Léger (THA)
0.9	Minor changes + C2 GIS updates	Antoine Léger (THA)
1	Chapter 1,4,5 and 6 rewriting	Antoine Léger (THA), Animesh Pathak (Inria), Valérie Issarny (Inria)

Document Review

Date	Version	Reviewer	Comment
12 February	1	V. Issarny (Inria)	Detail to be provided in the document for better understanding by the reader
16 February	2	N. Georgantas (Inria)	Editorial comments
24 February	3	V. Issarny (Inria)	-

Table of Contents

1 INTRODUCTION

The scope of WP6, as introduced in the Description of Work (DoW), is: "*to assess the CONNECT architecture and prototypes, as generated by WP1, against actual scenarios*". To this extent, the work performed within WP6 has concentrated on constructing such scenarios by identifying real systems that are available to the project members and that can provide a testbed against which to experiment with CONNECT technologies, defining in this way the actual scenarios that are expected in the Description of Work.

Specifically, during the third year, our goal has been to take into account the recommendations provided to us during the second review meeting while working towards clearly defining a rich scenario that includes several diverse Networked Systems (NS) and allows the consortium to assess the functionality of all the Connect enablers.

This deliverable specifically reports on the WP6 activities during Year 3, which have focused on the elaboration of the GMES scenario for experimenting CONNECT results and on defining our methodology for assessing CONNECT results against the project's goals stated in the DoW.

1.1 Summary of Achievements

In a nutshell, WP6 achievements over Year 3 relate to:

- Identification of the systems that will form the final evaluation platform;
- Definition of the Joint Forest-Fire Operation scenario that is part of the GMES use case;
- Specifications of the networked systems involved in the Joint Forest-Fire demonstration (interfaces, affordances and behavior);
- First implementations of representative networked systems;
- Elaboration of the assessment methodogy to be applied in the final year.

1.2 Second Review Recommendations

The project's 2nd review provided two main overall recommendations relevant to the work to be performed in Y3 within WP6:

- **Recommendation 4:** "*The Terrorist Alert scenario must have the capability to demonstrate all parts of the project including learning based synthesis, security and all the other enablers*."
- **Recommendation 6:** "*The consortium is asked to provide a description of a more systematic evaluation methodology that will be used to evaluate whether the project has reached its ambitious claims (such as supporting diversity, dependable, evolvable, highly heterogeneous, automated learning of models, adaptability, etc.)*."

The core of the work in WP6 during Y3, concentrated to a large extent on Recommendation 4 with the objective to introduce a GMES-related Scenario that not only illustrates and challenges all the CONNECT enablers developed within the project, but also shows the added-value of CONNECT in terms of integration and interoperability in a highly heterogeneous and dynamic context.

To address Recommendation 6, we have worked on an assessment plan to evaluate the various CONNECT enablers, as discussed in detail in Chapter 5.

In addition to the above, the following recommendations were specifically made for WP6 work:

1. "*At the last review, we recommended that this WP selects one scenario that could work across the whole project. The WP has now defined a new scenario – the Terrorist Alert scenario – that includes a wide set of interaction mechanisms that must be mediated*

*by Connect and in addition requires attention to non-functional properties such as performance, location and access contro*l."

2. "*The reviewers recommend that to make impact, the demonstration should deal with live video streams rather than copying video files. The reviewers recognized that it would be necessary to use pre-compiled translators for different video stream formats selected rather than synthesized by Connect. The scenario can still accommodate synthesis in the context of connection management*."
3. "*So far only a discovery Enabler has been implemented in the testbed scenario, learning, monitoring, and synthesis remain to be addressed in the upcoming year*."

To address Comment 1, the GMES scenario now includes a wide set of interaction mechanisms, and exhibits non-functional aspects such as location and access control.

To address Comment 2, the scenario now includes two networked systems dealing with live video streams – cameras (both ground based and airborne) that relay live video streams, as well as a live multimedia communication between fire-fighters.

To address Comment 3, this deliverable includes a report on the development status of all the CONNECT enablers, which will be experimented using the networked systems of the GMES-based scenario and their connection.

1.3 Challenges for Year 3

The main challenges for Year 3 have been: (i) the definition of the GMES scenario that involves all the CONNECT enablers, while remaining realistic, as well as (ii) the implementation of the various networked systems involved in the scenario.

The scenario is in particular expected to illustrate the usage of various interaction paradigms (e.g., RPC, Publish /Subscribe, Messaging and Shared Space), thereby illustrating on-the-fly mediation of protocols from the application down to the middleware layers.

1.4 Achievements in Year 3

The main goal of Year 3 has been to build the GMES use case demonstrator, from the results of the last year demonstration on "Terrorism Alert". Along these lines, the main achievements of WP6 in Year 3 have been the following:

- The definition of the GMES scenario that satisfies the requirements expressed above in Section 1.3. Within the demonstration, independent systems are available; some are from industrial partners others are from the public market (UAV, Cameras, weather station, etc.). Furthermore, the demonstration shows heterogeneous systems exchanging data using various data format and transports protocols.
- The scenario now includes the connection of more than two systems.
- The scenario integrates the Fire-fighter VideoConference and live video streaming adaptation issues.
- The first version of representative networked systems for the identified data exchange patterns have been implemented:
    - Client/Server: cameras PTZ, Weather service, UAV.
    - Publish /Subscribe: Positioning System A.
    - Message: Positioning System B.
    - Shared memory: weather station.

In addition, as already mentioned, we have elaborated the assessment methodology to be applied to assess CONNECT results against the project's goals stated in the DoW.

1.5 Structure of the Deliverable

The rest of the deliverable is organized as follows:

- Section 2 highlights the GMES scenario and the actors and systems involved;
- Section 3 addresses the description of each networked system to implement to enable the scenario, introducing the interfaces, the affordance and the behaviour (in BPEL) of each NS;
- Section 4 provides a global picture of the demonstrator.
- Section 5 concentrates on the assessment of CONNECT solutions against the project's objectives stated in the DoW. The proposed assessment spans (i) the project's overall objective of enabling on-the-fly interoperability among heterogeneous networked systems as well as (ii) the project's specific objectives related to the foundations and associated enablers to be elaborated for learning and reasoning about the interaction behaviours of networked systems and for synthesizing mediators so as to make systems interoperate.
- The Annex provides the SAWSDL (semantically-annotated WSDL) for each networked systems described in Section 3, along with the OWL ontology that is used to annotate the interfaces and descriptions.

2 EXPERIMENT: THE GMES USE CASE

The GMES use case will set the ground for the assessment of CONNECT in the System of Systems area. It will challenge the overall approach towards supporting large scale and highly heterogeneous applications.

2.1 GMES Context

GMES (Global Monitoring for Environment and Security) is the European Programme for the establishment of a European capacity for Earth Observation started in 1998. The services provided by GMES address six main thematic areas: land monitoring, marine environment monitoring, atmosphere monitoring, emergency management, security and climate change.

The emergency management service directs efforts towards a wide range of emergency situations; in particular it covers different catastrophic circumstances: floods, forest fires, landslides, earthquakes and volcanic eruptions and humanitarian crises.

Within CONNECT, we concentrate on the Forest fire situation.

2.2 Scenario Description: Joint Forest-Fire Operation

Fires are responsible for the destruction of huge areas of forests every summer in Europe. These disasters are very damaging to the ecosystem.

Due to climate change (global warming, dryness), the recent forest fires have been wider (Greece, 2007) and thus required more fighting resources, and even cross-state/country (Greece-Albania, 2011). However, each European nation has its own organization, resources and strategies to handle such fires so when a neighbour country provides support, it is often with different means, resources or protocols.

For example in France, the French Civil Protection is an organization in charge of emergencies and disasters management; in particular, the forest fires. It regroups multiple organizations such as police, fire-fighters, assistance, etc.

We detail below the French Civil Protection structure shown in Figure 1. During a forest fire, the French fire-fighters are organized according a predefined hierarchy. The smallest fire-fighting unit is an “attack group”. An attack group is composed of five vehicles, each transporting four fire-fighters (one driver and three terrain fire-fighters). An attack group is led by a “chief of group”. Three attack groups can be clustered in a “column”, led by a chief of column. When deployed on a forest fire, the columns are under the command of a “*Commandant des opérations de secours*” (Commander of Emergencies Operations) abbreviated as COS. The COS is generally located in a vehicle called “Poste de Commandement” (Field Command Centre) called Command and Control cell in this document. This vehicle is equipped with computers and communication means. The COS is in relationship with the « *Centre Opérationnel Départemental d'Incendie et de Secours* » (Departmental Operational Centre for Fires and Emergencies) abbreviated as CODIS. The CODIS is an organization usually located in department’s prefecture. It is linked to an emergency call centre receiving alerts for emergencies such as forest fires. The regional administrator called “*Préfet*” manages on a higher level the actions of CODIS especially in case of wide disasters

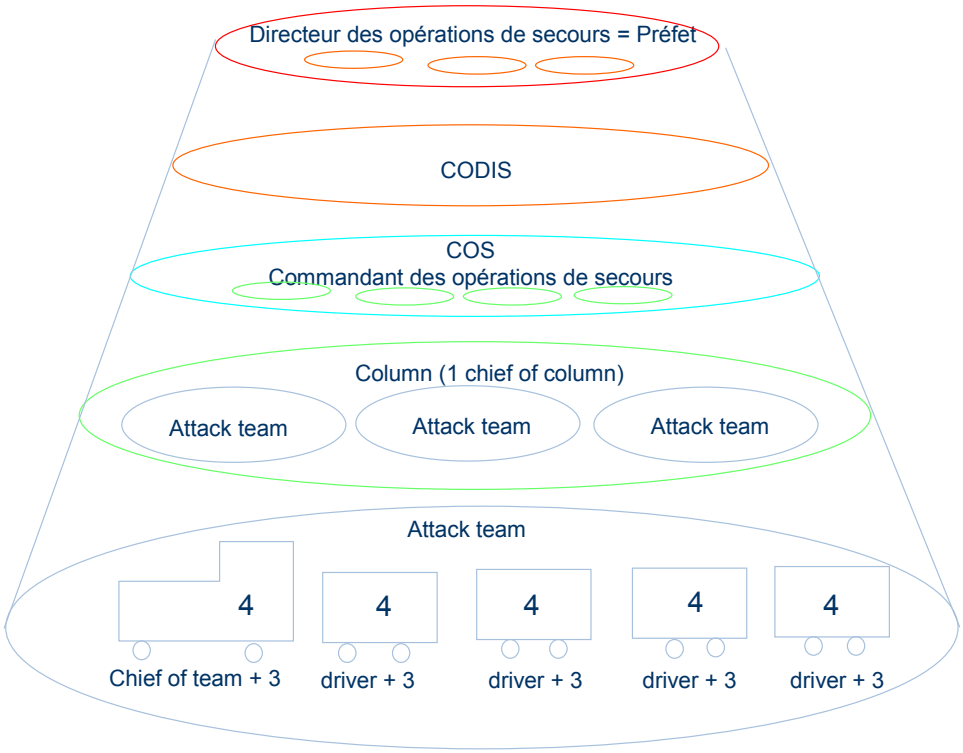

Figure 1: Decision hierarchy for forest fire management

2.2.1 Scope

The Joint Forest-Fire Operation scenario will demonstrate the functionalities of the CONNECT architecture and its added value in the monitoring and the management of a forest fire, by enabling the CONNECTion of the heterogeneous NSs involved.

Precisely, the scenario will demonstrate the capabilities of CONNECT to discover and synthesize CONNECTors in a crisis situation. The global efficiency improvement will be achieved through multi-source data combination, with:

- An easy integration of new sensors and services in operational systems,
- A facilitated discovery and access to sensors and their data, and services.

2.2.2 Scenario details

The scenario illustrates the management of forest-fire, close to a border village and a factory between country A and country B. Forest monitoring and forest fire management in the country A are the responsibility of the Country A Command and Control fire operations center (C2-A). For example in France, the CODIS, led by a professional fire-fighter, is the regional authority in charge of coordination of operational forces during fires and disasters for one French department.

The scenario addresses two phases:

- **Forest fire crisis**: this phase addresses the reporting of alerts and alarms to decision makers and operational forces, the operational deployment of intervention forces and sensors, and the fire-fighting period. During the fire-fighting period, some specific strategies are performed and analyzed, including the identification and connection to available information sources like the unmanned ground vehicles (UGV) and IR video camera from the factory or the Traffic monitoring camera of the village. In case the fire

goes wider and there is a direct threat to the village and the factory, country A asks for support of country B.

- **Reinforcement integration**: Country B provides to country A an unmanned aerial vehicle (UAV) to get a better view of the fire front close to the village in order to be able to proceed to its evacuation in time. Country B also grants access to its weather forecast service and the detachment of a full fire fighter squad (Squad 2) on fire front under the C2-A command.

We detail the above 2 phases in the next sections.

2.2.3 Crisis phase

The crisis phase starts when a fire is reported, leading to a fire-fighting strategy being defined and fire-fighters attacking the fire. The "Country A"'s Command and Control fire operations center (C2-A) is in charge of the management of the forest fire crisis. This phase further involves a number of sensors and human actors, as listed below:

- **Sensors involved:**
  - UGV:
    - Dynamic Areas of interest like threatened peri-urban zones,
    - High resolution images of dangerous areas without direct exposure of fire-fighters.
  - Positioning Sensors for:
    - Locating fire-fighters and resources deployed around the forest fire
    - Positions are available via a dedicated Position Service
  - Video Cameras for surveillance of remote areas as:
    - Evacuated zones
    - Peri-urban zones threatened by a fire
    - Strategic traffic points

  These cameras can be either fixed or controlled (Pan-Tilt-Zoom, or PTZ).
  - A mobile weather station for local weather information
- **Human actors involved:**
  - Resources from C2-A staff and UGV staff ,
  - Actors on the theatre of operations:
    - The chief in the Command and Control cell,
    - Operators in the Command and Control cell,
    - Fire-fighters Squad 1
    - All relevant administrative bodies (local, regional, etc.).

The crisis phase is divided into four steps, i.e., alert, deployment, fire-fighting, and fire evolution, as detailed hereafter:

- ***Step 1.1: Alert (Figure 2)***

  The alert step addresses the alarm and its reporting to competent authorities and suitable operational forces

  The crisis phase is then entered when a fire is reported by existing means of detection (e.g. human detection from watchtowers spread throughout the forest), or by alarm by members of the public.

  Using the resources available in the C2-A:

  - Alerts are registered and located.
  - Forecasts of possible evolution of the fire are done by using local weather information and models.
  - The C2-A assesses the situation and alerts relevant actors.
  - A fire-fighting strategy is defined and coordinated.

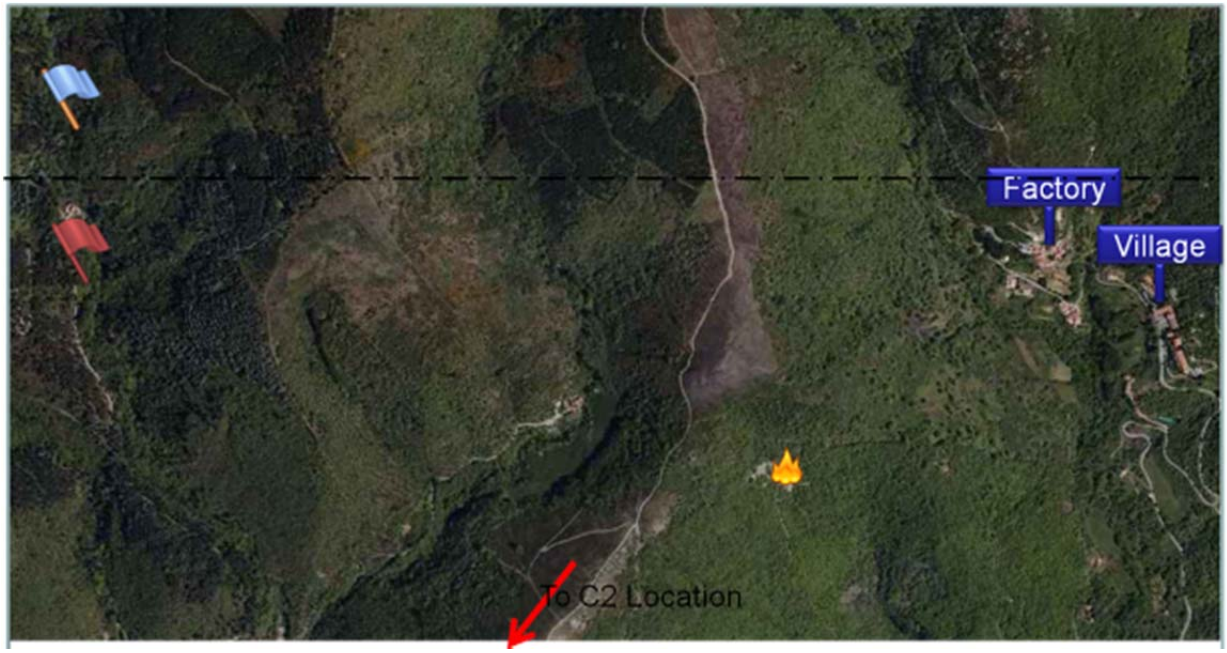

Figure 2: Overview at Step 1.1

During this phase, potentially "hot" areas are determined by forecasting the fire behaviour according a weather model, the typology of the terrain, the vegetation and the type of area (for example: forest, housing, factories, etc.).

These simulations give the principal axis of propagation of the fire, and this axis is used to define two sectors (right and left sides of the axis). Non-burned areas where fire could propagate are identified. The C2-A inform operational forces of the threatened zones. The foreseen meteorological conditions concerning the fire area will be confirmed by fire-fighters on their arrival using a mobile weather station.

- ***Step 1.2: Deployment of sensors and operational forces (Figure 3)***

  In the deployment step, the operational forces arrive on site. Several sensors are deployed and/or set in an active mode (switched on) to enhance situation understanding and assessment. This includes the discovery and the connection to available resources even those that weren't foreseen (the factory resources are exposed via the discovery enabler)

  According to the fire-fighting strategy defined during the alert phase, fire-fighting means, infrastructure and sensors are deployed on site and made accessible following these steps:

  1- On arrival at the crisis area, fire-fighters are deployed according to the chosen fire-fighting strategy. A Command and Control cell is deployed for on-site operational coordination. This cell is provided with an application displaying the current operational situation. An operator and a local commander are in the Command and Control cell. The local commander (COS) communicates with operational forces fighting the fire. Two communications means are available: the classical radio or a live videoconference system using which the C2 is able to initiate multimedia communications with squad leaders requiring audio and/or video streams setup and adaptation across heterogeneous mobile devices. Supported interaction models will include Facetime-like and push2talk (walkie-talkie) communication.

2- Operators deploy in-situ sensors and on-site facilities.
Some of the fire-fighters are equipped with a GPS positioning system.
For all areas of interests, a means to get information is identified / discovered or deployed.
   a. Here a wireless camera is deployed in AOI 1 – “the crossroad”.
   b. A local mobile meteorological station is deployed in the vicinity of the affected region.

Once the whole system is operational, information is displayed and used at the Command and Control cell for the selection of the fire-fighting strategy.

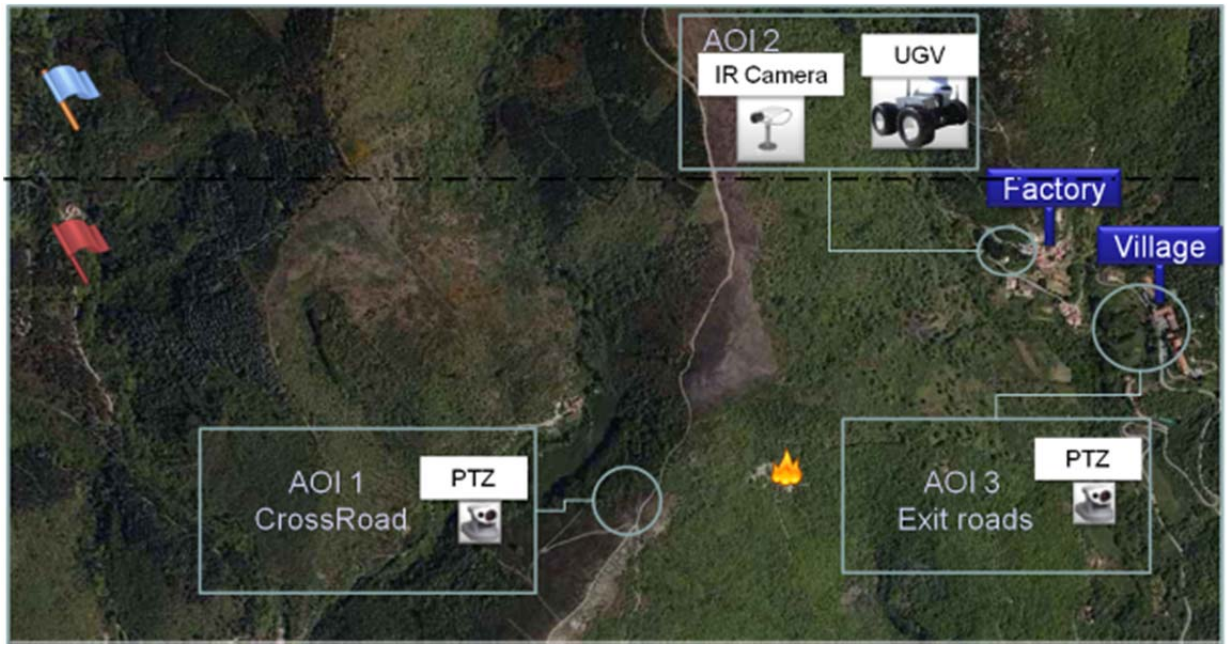

Figure 3: Overview at Step 1.2

- ***Step 1.3: Fire-fighting phase (Figure 4)***

  During the fire-fighting step, fire-fighters are in action against the fire and previously deployed sensors provide data about observed phenomena.

  In particular, the Command and Control cell continuously receives all the necessary information to adapt the fire-fighting strategy according to the situation evolution (images/video from cameras, positions of the fire-fighters, terrain information from the fire-fighters, local weather forecast, wind intensity and direction, traffic diversions).

  Several actions are performed during the fire-fighting phase:

  o *Command and control*

    The Command and Control Cell has a large amount of information at its disposal through a GIS in order to manage the crisis. The following activities are executed:

    - coordination of the operations in the fire perimeter,
    - exchange of information between C2 and deployed teams,
    - access to data from sensors,
    - display of information relative to crisis situation on a cartographic display (using symbols, images or video),
    - building of global operational view,
    - monitoring & control of UGV / PTZ Camera

- *Surveillance of areas interest*

  For all areas of interests, a means to get information is identified / discovered or deployed. These areas can be, for example, housing areas, camping-sites or strategic points such as junction crossroads to monitor trafficking trucks etc. Cameras can support this surveillance. Cameras help to reduce the number of fire-fighters dedicated to surveillance and as a consequence help to fight the fire. For instance, considering the scene of Figure 4, we have:

  a. For AOI 1, “the crossroad”, a wireless PTZ camera is deployed.
  b. For AOI 2, “the factory”, the exposed resources from the factory are discovered, i.e., UGV and IR Fixed camera
  c. For AOI 3, “the Village exit roads”, the Traffic monitoring cameras are discovered, i.e., 1 PTZ and 1 Fixed

Figure 4: Overview at Step 1.3

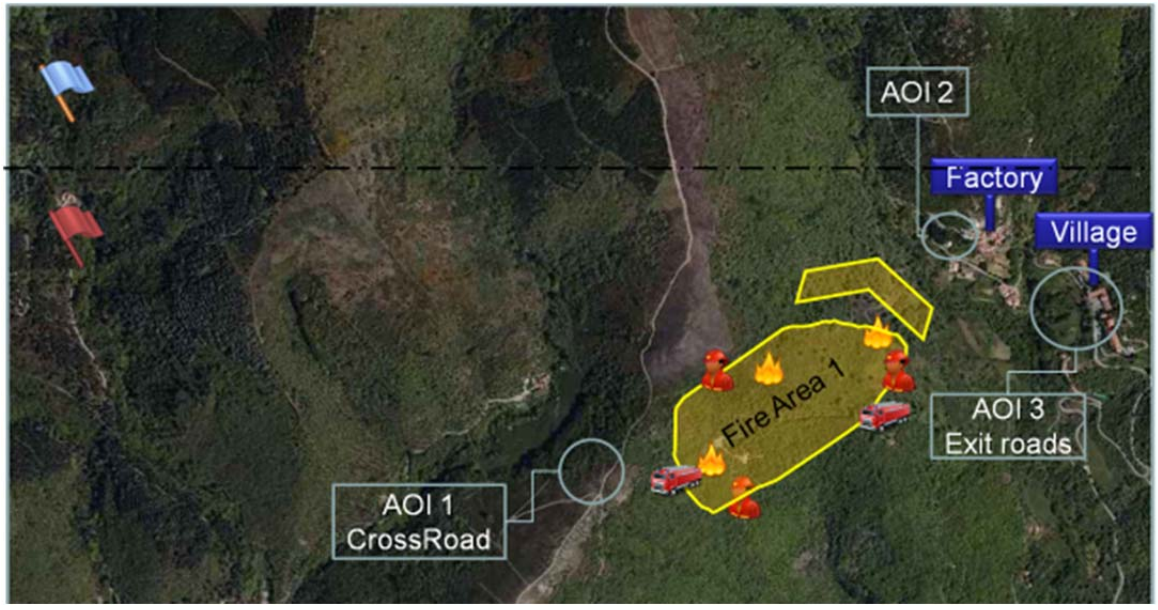

Figure 5: Overview at Step 1.4

- ***Step 1.4: Fire evolution phase (Figure 5)***

  The fire evolution requests more resources and a closer survey of the village, which leads to request for support to country B. As a result, Country B shall initiate the deployment of supporting resources (UAV, Weather service, Fire-fighters Squad 2).

  The fire evolution and forecast shows that the fire will spread in direction of the village in a second fire area (Fire Area 2). This situation requests more fighting resources and the evacuation of the factory next to the village. A closer survey of the factory is also needed in order to anticipate village evacuation. So the decision is made to ask for support from Country B.

2.2.4 Reinforcements integration phase

The reinforcement phase starts when the support resources from country B are deployed; as a result, the fire-fighting strategy is updated and fire-fighters attack the fire on both fire areas. The “Country A”’s Command and Control fire operations center (C2-A) is in charge of the management of the forest fire crisis. This phase further involves a number of sensors and human actors, as listed below:

- **Sensors involved**
  - UAV:
    - General view of the situation
    - Detection and time evolution of the fire frontline
    - Areas of interest like threatened peri-urban zones,
    - High resolution images of limited areas where fire fighters are in trouble
  - UGV:
    - Dynamic Areas of interest like threatened peri-urban zones,
    - High resolution images of dangerous areas without direct exposition of fire-fighters.
  - Positioning Sensors for:
    - Locating fire-fighters and resources deployed around the forest fire
    - Positions are available via a dedicated Position Service
    - Country B Positions Service for country resources location
  - Video Cameras for example for surveillance of remote areas as:
    - Evacuated zones
    - Peri-urban zones threatened by a fire
    - Strategic traffic points

    These cameras can be either fixed or controlled (PTZ)
  - A mobile weather station for local weather information
  - A weather service with regional and national coverage
- **Human actors involved**
  - Resources from the crisis phase (C2 staff and UGV staff)
  - Actors on the theatre of operations:
    - The chief in the Command and Control cell,
    - Operators in the Command and Control cell,
    - Fire-fighters Squad 1
    - Fire-fighters Squad 2
    - All relevant administrative bodies (local, regional, etc.).

The Reinforcement integration phase is then divided into 4 steps:

- ***Step 2.1: Discovery of new resources***

  Once deployed on the operation theatre, the resources from the Country B are seen by the C2 Operator as resources available and usable during the Fire-fighting phase.

- ***Step 2.2: Connection to Country B Position Service***

The C2 operator initiates the subscription to the Country B Position service and starts receiving positions of the Country B resources.

- ***Step 2.3: Consume the Country B Weather service***

  In order to update the environmental picture of the operation theatre, the C2-A operator starts consuming the Country B Weather Service and receives current weather information and displays them on screen in a dedicated layer. To refine the knowledge about fire evolution, he now needs to access to weather forecast and performs the login steps in order to have a granted access to such information and then displays them in a dedicated layer.

- ***Step 2.4: Fire-fighting phase (Figure 6)***

  Step 2.4 is as Step 1.3 on the Country B side.

  This step is indeed similar to the earlier C&C action discussed under Step 1.3 (before the help given by country B), but with the involvement of additional services and devices provided by country B, i.e., weather service and UAVs.

  In particular, in the fire survey, the UAV is positioned above the area where the fire is active. The survey is performed according to a predefined flying pattern. Depending on the video analysis, the operators in the Command and Control cell have the possibility to change its flying pattern using one of the predefined flying patterns.

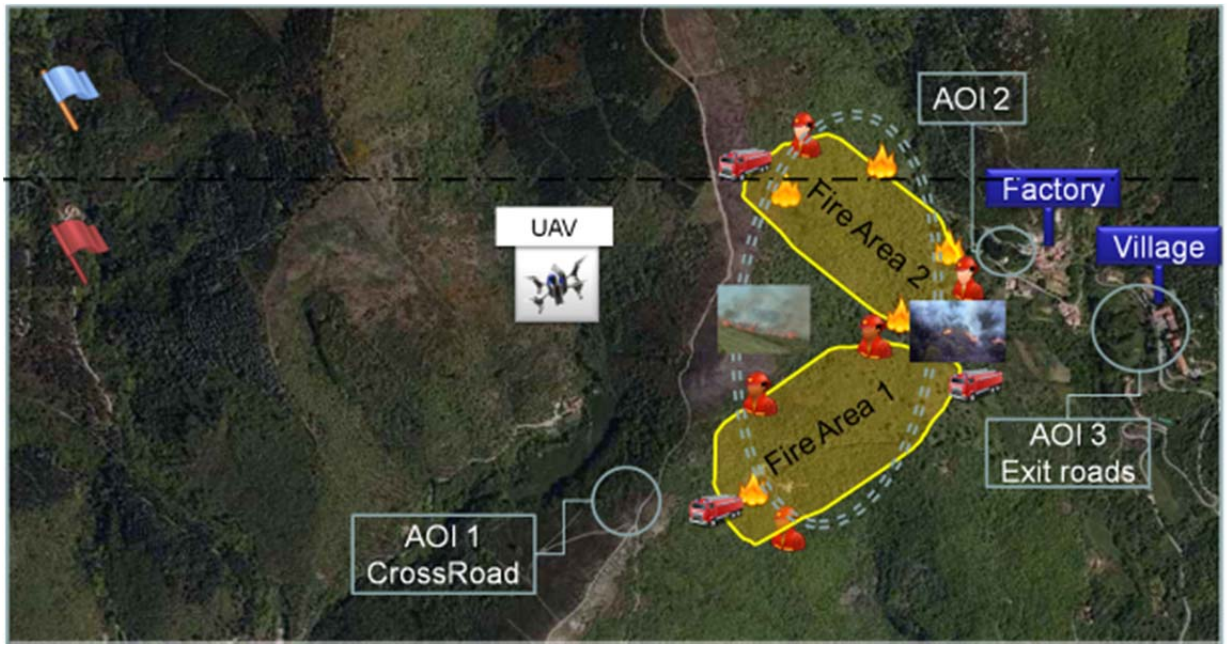

Figure 6: Overview at Step 2.4

3 NETWORKED SYSTEMS

This chapter provides the list of the networked systems (NS) involved within the GMES-based scenario discussed in the previous chapter, i.e.:

- **NS 1.1** - UAV
- **NS 1.2** - UAV Camera
- **NS 2** - UGV
- **NS 3.1** - Camera Fixed
- **NS 3.2** - Camera Mobile Main
- **NS 3.3** - Camera Mobile Patrol
- **NS 4** - C2 GIS
- **NS 5** - Mobile Weather Station
- **NS 6** - Weather Service
- **NS 7.1** - Positioning System – Country A
- **NS 7.2** - Positioning System – Country B
- **NS 8** - Firefighter Live Multimedia Communication

For each NS, we provide:

- The system's main function and underlying interaction and discovery protocols;
- The system's syntactic interface, while the associated SAWSDL file is provided in Annex;
- The system's affordances (offered and consumed services);
- The system's behaviour; and
- The system's technical specifications.

As detailed in WP1-related deliverables, the semantics of the NSs' operations and affordances are specified using ontology concepts. The definition of the domain-specific ontology used in our GMES-based scenario is given in Appendix (see Section 7.13). As an illustration, Figure 7 depicts the hierarchical graph of concepts used for the definition of affordances, while Figure 8 gives the graph associated with GMES operations.

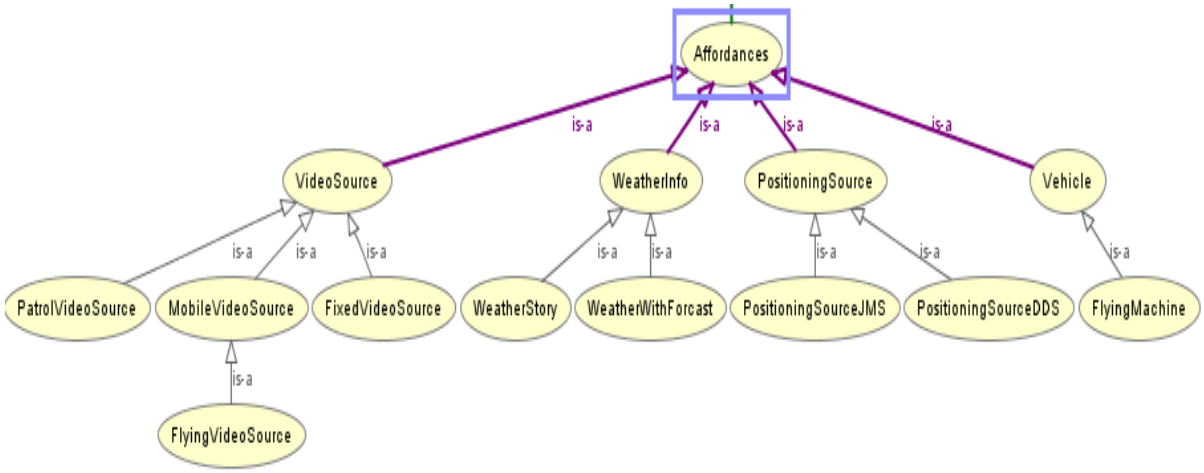

Figure 7: Concepts used in affordances

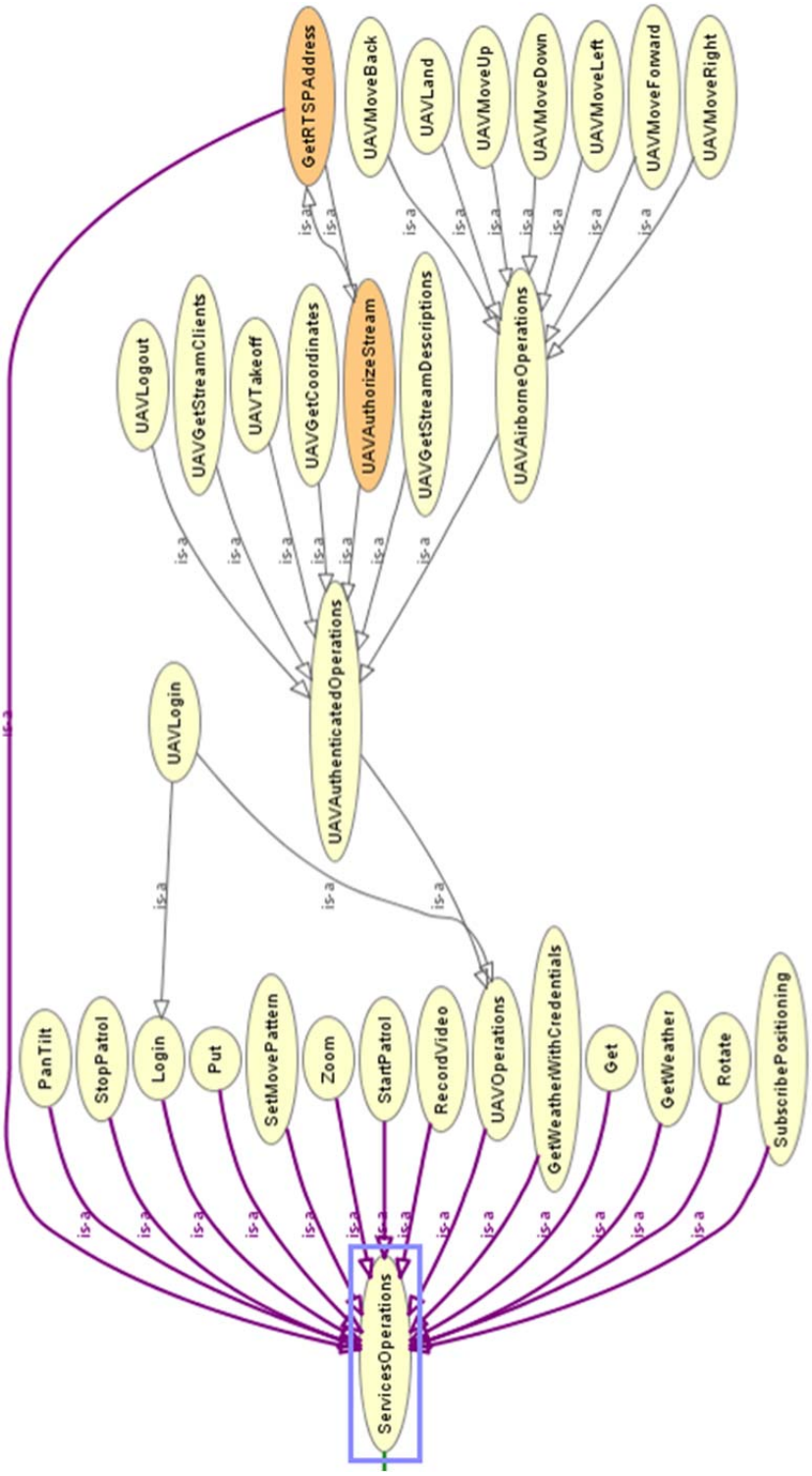

Figure 8: Concepts used in SAWSDL

3.1 NS 1.1 - UAV

The UAV system is a flying mobile platform, hosting a UAV Camera. It provides a SOAP Web service for its control operations, and uses CDP as Discovery Protocol.

3.1.1 Interfaces

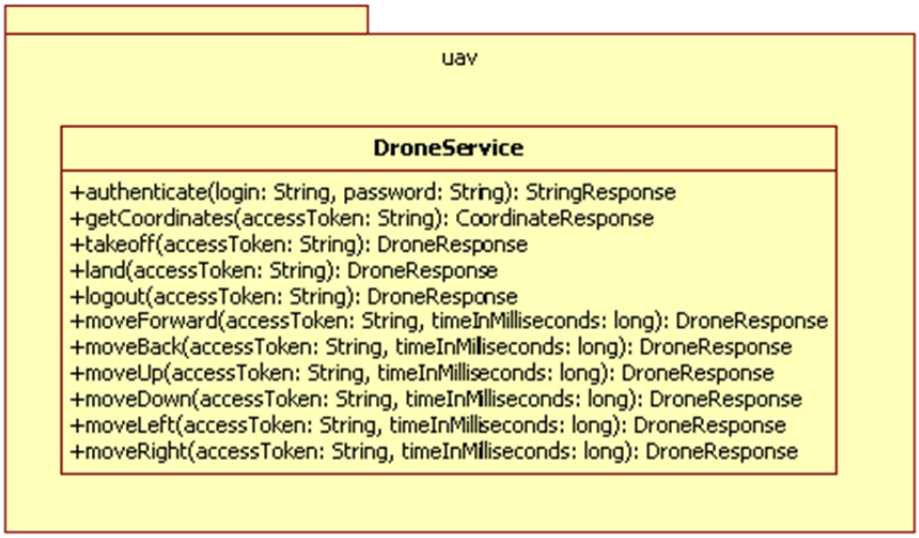

Figure 9: UAV Interface

authenticate*(login: String, password: String): StringResponse* – authenticates with a login and password, returns an access token to be used for all other commands

getCoordinates*(accessToken: String): CoordinateResponse* – gets current coordinates of UAV, returns a 6D value including roll, pitch, and yaw in addition to X,Y, and Z

takeoff*(accessToken: String): DroneResponse* – orders the UAV to take off

land*(accessToken: String): DroneResponse* – orders the UAV to land

logout*(accessToken: String): DroneResponse* – invalidates the access token

moveForward*(accessToken: String, timeInMilliseconds: long): DroneResponse* – self explanatory

moveBack*(accessToken: String, timeInMilliseconds: long): DroneResponse* – self explanatory

moveUp*(accessToken: String, timeInMilliseconds: long): DroneResponse* – self explanatory

moveDown*(accessToken: String, timeInMilliseconds: long): DroneResponse* – self explanatory

moveLeft*(accessToken: String, timeInMilliseconds: long): DroneResponse* – self explanatory

moveRight*(accessToken: String, timeInMilliseconds: long): DroneResponse* – self explanatory

3.1.2 Affordance

The UAV's affordance is about providing a vehicle which can move in 3D space. The affordance itself is declared as a subclass of the general "Vehicle" affordance.

```
<Affordances name="FlyingMachine">
        <Affordance name="FlyingMachineAff" kind="provided">
                <FunctionalConcept>http://www.connect.com/ontology/media#FlyingMachine
</FunctionalConcept>
                <Inputs></Inputs>
                <Outputs> </Outputs>
        </Affordance>
</Affordances>
```

3.1.3 Behaviour

The behaviour of the UAV affordance is illustrated in ***Figure 10***. The user first needs to authenticate with the service. After authenticating, the user can get the coordinates of the UAV, or order it to move left, right, front, back, up or down, or land. Note that the movements

are contingent on first invoking the command for the UAV to take off. Each of the movement operations takes a time duration as an argument, which controls how far the UAV will go.

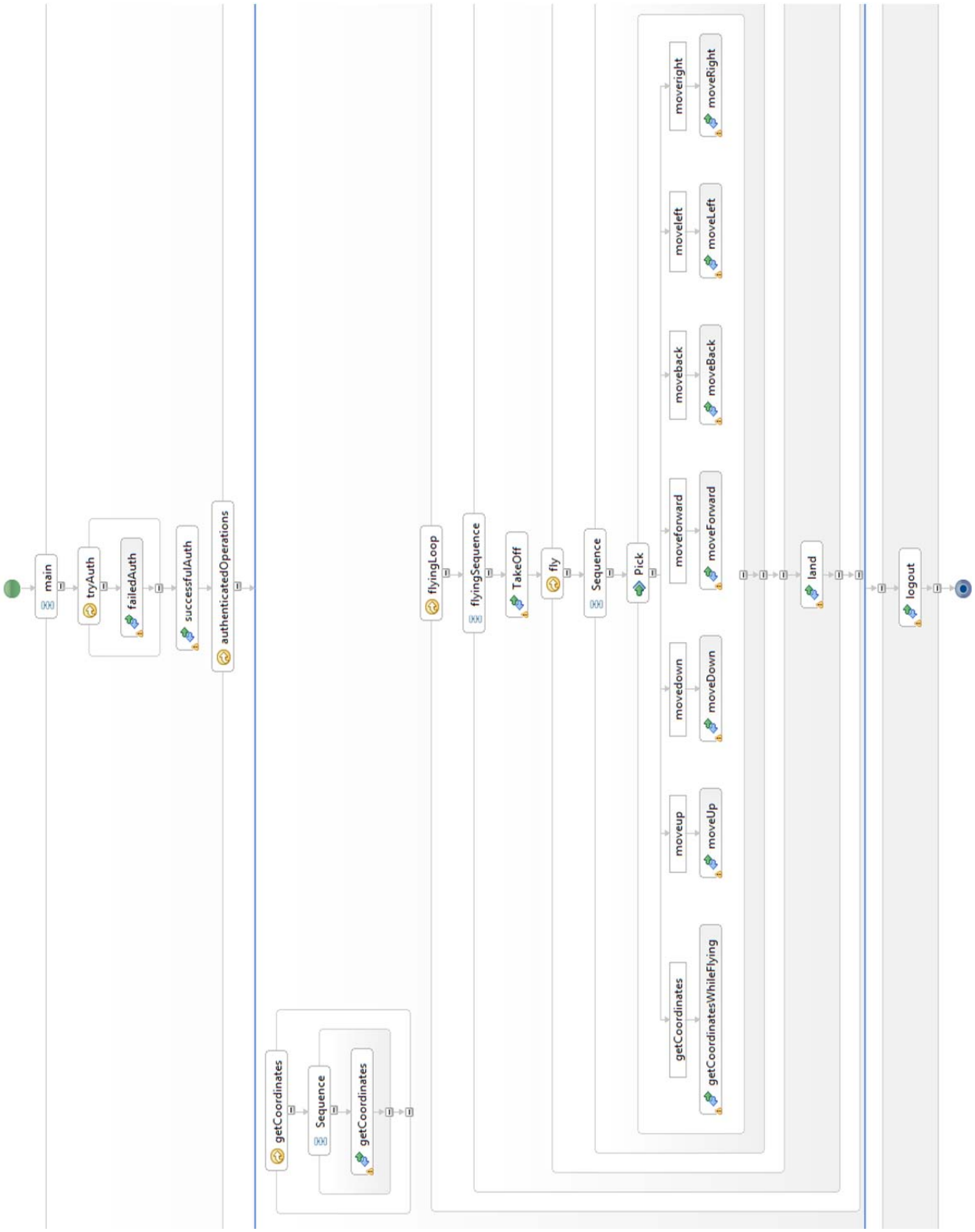

Figure 10: Behaviour of UAV

3.1.4 Technical specifications

	**AR.Drone Parrot**
	MOTORS AND ENERGY - 4 brushless motors, (35,000 rpm, power: 15W) - Lithium polymer battery (3 cells, 11,1V, 1000 mAh) - Discharge capacity: 10C - Battery charging time: 90 minutes
	FRONT CAMERA: WIDE ANGLE CAMERA - 93° wide-angle diagonal lens camera, CMOS sensor - Encoding and live streaming of images on iPhone - Camera resolution 640x480 pixels (VGA) - Video feedback on the iPod touch /iPhone screen
	ULTRASOUND ALTIMETER - Emission frequency: - 40kHz - Range 6 meters (19.7 ft) Vertical stabilization
	VERTICAL CAMERA: HIGH SPEED CAMERA - 64° diagonal lens, CMOS sensor - Video frequency: 60 fps - Allows stabilization even with a light wind
	EMBEDDED COMPUTER SYSTEM - ARM9 468 MHz - DDR 128 Mbyte at 200MHz - Wifi b/g - USB high speed - Linux OS
	INERTIAL GUIDANCE SYSTEMS WITH MEMS - 3 axis accelerometer - 2 axis gyrometer - 1 axis yaw precision gyrometer
	SPECIFICATIONS - **Running speed:** 5 m/s; 18 km/h (16.4 ft per second; 11.2 miles per hour) - **Weight:** - 380 g with outdoor hull (0.8 pounds) - 420 g with indoor hull (0.9 pounds) - **Flying time:** about 12 minutes
	SAFETY SYSTEM - EPP hull for indoor flight - Automatic locking of propellers in the event of contact - UL2054 battery - Control interface with emergency button to stop the motors
	DIMENSIONS - **With hull:** 52,5x51,5cm (20.7 inches x 20.3 inches) - **Without hull:** 45x29cm (17.7 inches x 11.4 inches)

Figure 11: AR.Drone Parrot Technical details

3.2 NS 1.2 - UAV Camera

The Uav camera is a video camera embedded in a UAV, offering an RTSP video stream. It uses CDP as Discovery Protocol.

3.2.1 Interfaces

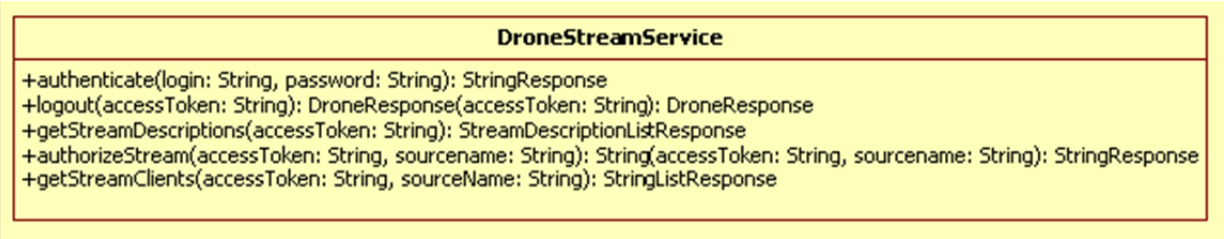

Figure 12: UAV Camera Interface

authenticate*(login: String, password: String): StringResponse* - authenticates with a login and password, returns an access token to be used for all other commands

logout*(accessToken: String): DroneResponse* – invalidates the access token

getStreamDescriptions*(accessToken: String): StreamDescriptionListResponse* - returns a list of StreamDescriptions describing the nature of the streams provided by the UAV

authorizeStream*(accessToken: String, sourcename: String): StringResponse* - authorizes a stream session to the current user

getStreamClients*(accessToken:String sourceName: String)*: StringListResponse - gets the list of clients connected to a particular streaming source

3.2.2 Affordance

The affordance of the UAV Camera is to provide one or more live video streams from cameras attached to a flying vehicle. The concept of a *FlyingVideoSource* is a subclass of a *VideoSource*, as defined in the domain ontology.

```
<Affordances name="FlyingCamera">
        <Affordance name="FlyingCameraAff" kind="provided">
                <FunctionalConcept>http://www.connect.com/ontology/media#FlyingVideoSource
</FunctionalConcept>
                <Inputs></Inputs>
                <Outputs> </Outputs>
        </Affordance>
</Affordances>
```

3.2.3 Behaviour

The behaviour of UAV Camera is illustrated in **Figure 13**. First, the system waits for an authentication message. Then, the usage of the UAV Camera can be monitored using the "*getStreamClients*" command. This request can be sent at any time after successful authentication. At this time, the NS can also receive a "*getStreamDescriptions*" request. The response to "*getStreamDescriptions*" contains a list of VideoSource descriptions. Such descriptions are required, since UAVs can be equipped with multiple video sources. The video source can be either live (fixed or mobile cameras) or prerecorded by the on-board cameras. The UAV can manage only a restricted number of clients at a time because of quality of service requirements. In order to enforce the number of clients connecting to a *VideoSource*, a client must send an "*authorizeStream*" request prior to attempting to connect to a stream. Any session must end with a logout message. If a client is still connected to a *VideoSource* after a "logout" request was sent, the UAV Camera will kill its connection.

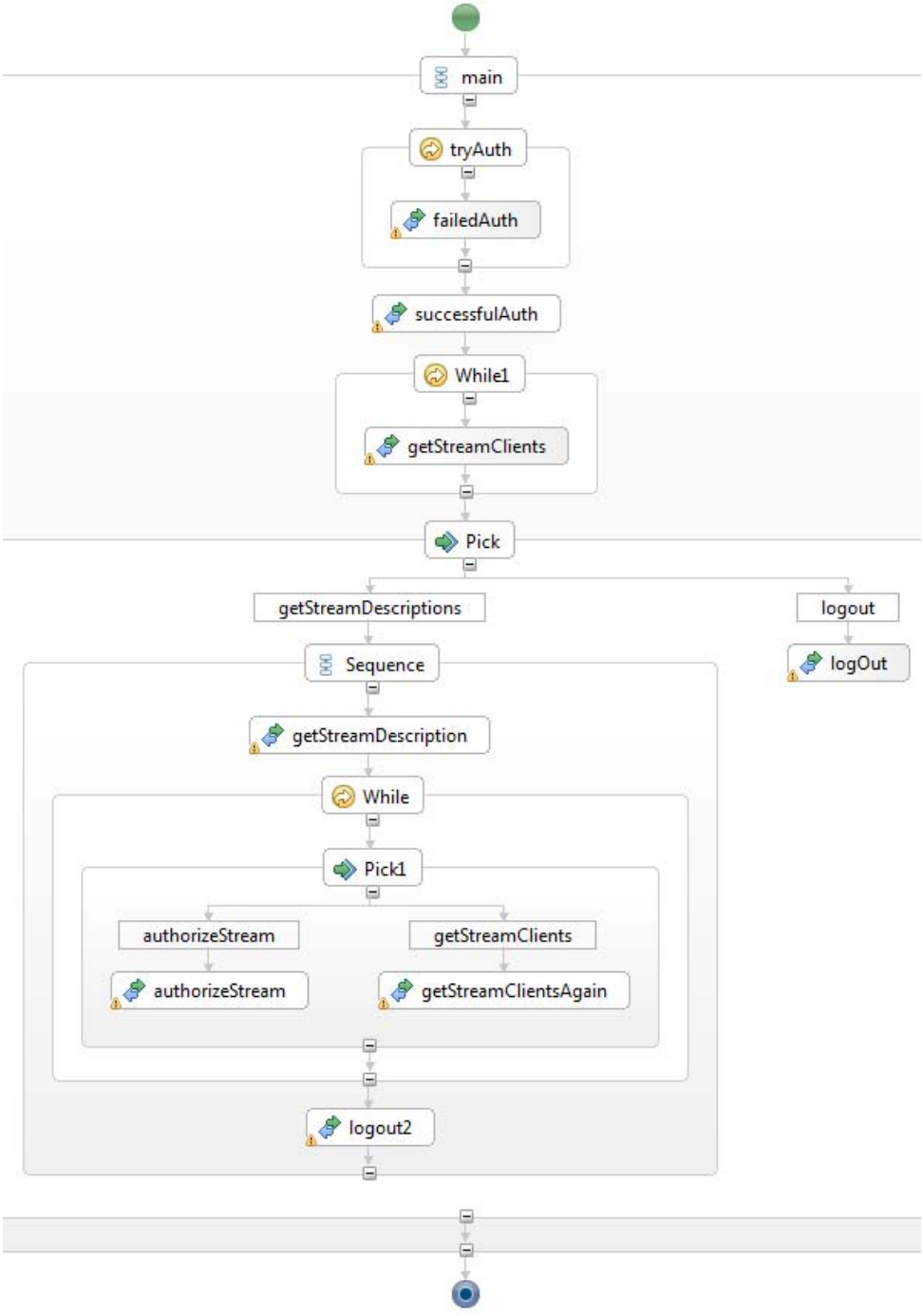

Figure 13: UAV camera Behaviour

3.2.4 Technical Specifications

See the table in Section 3.1.4

3.3 NS 2 - UGV

UGV provides access to a video stream issued from a video camera embedded on a ground mobile platform, controlled using remote procedure call (RPC) over HTTP. It uses CDP as Discovery Protocol.

3.3.1 Interfaces

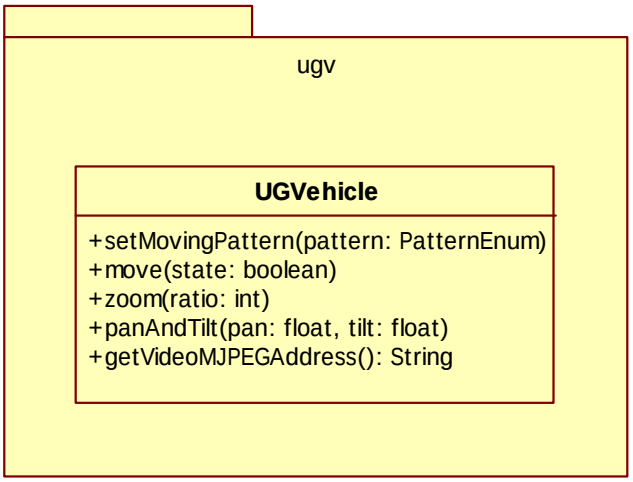

Figure 14: UGV interface

setMovingPattern*(PatternEnum pattern)* - sets the pattern of the vehicle movement. For moving, use the move operation. *pattern* : can take different values among CIRCLE SQUARE DIAMOND

move*(boolean state)* - makes the vehicule to move following its defined pattern or to stop. state: true or false

zoom*(int ratio)* - sends a zoom command to the camera. *ratio* defines the relative zoom value

panAndTilt*(float pan, float tilt)* - sends a pan/tilt command to the camera. *pan*: defines the relative pan value. *tilt*: defines the relative tilt value

getVideoMJPEGAddress*()* - returns the address of the video MJPEG flux of the camera

3.3.2 Affordance

The UGV possesses two affordances, the first one with the functional concept "*MobileVideoSource*" that describes the capacity of its embedded mobile camera. The second is the concept "*Vehicle*" that represents its capacity to move.

```
<?xml version="1.0" encoding="UTF-8"?>
<Affordances name="UGVehicle">
	<Affordance name="UGVehicleAff1" kind="provided">

	<FunctionalConcept>http://www.connect.com/ontology/media#MobileVideoSource</FunctionalCo
ncept>
		<Inputs>
			<Input>http://www.connect.com/ontology/media#Void</Input>
		</Inputs>
		<Outputs>
			<Output>http://www.connect.com/ontology/media#Void</Output>
		</Outputs>
	</Affordance>
	<Affordance name="UGVehicleAff2" kind="provided">

	<FunctionalConcept>http://www.connect.com/ontology/media#Vehicle</FunctionalConcept>
		<Inputs>
			<Input>http://www.connect.com/ontology/media#Void</Input>
		</Inputs>
		<Outputs>
			<Output>http://www.connect.com/ontology/media#Void</Output>
		</Outputs>
	</Affordance>
</Affordances>
```

3.3.3 Behaviour

The following BPEL explains the sequence of operation permitted on the UGV. To move the UGV, the operation "*setMovingPattern*" must be called before the operation "*move*". The other operations '"*zoom*", "*pantAndTilt*", "*getVideoMJPEGAddress*" target the embedded camera, and can be called anytime.

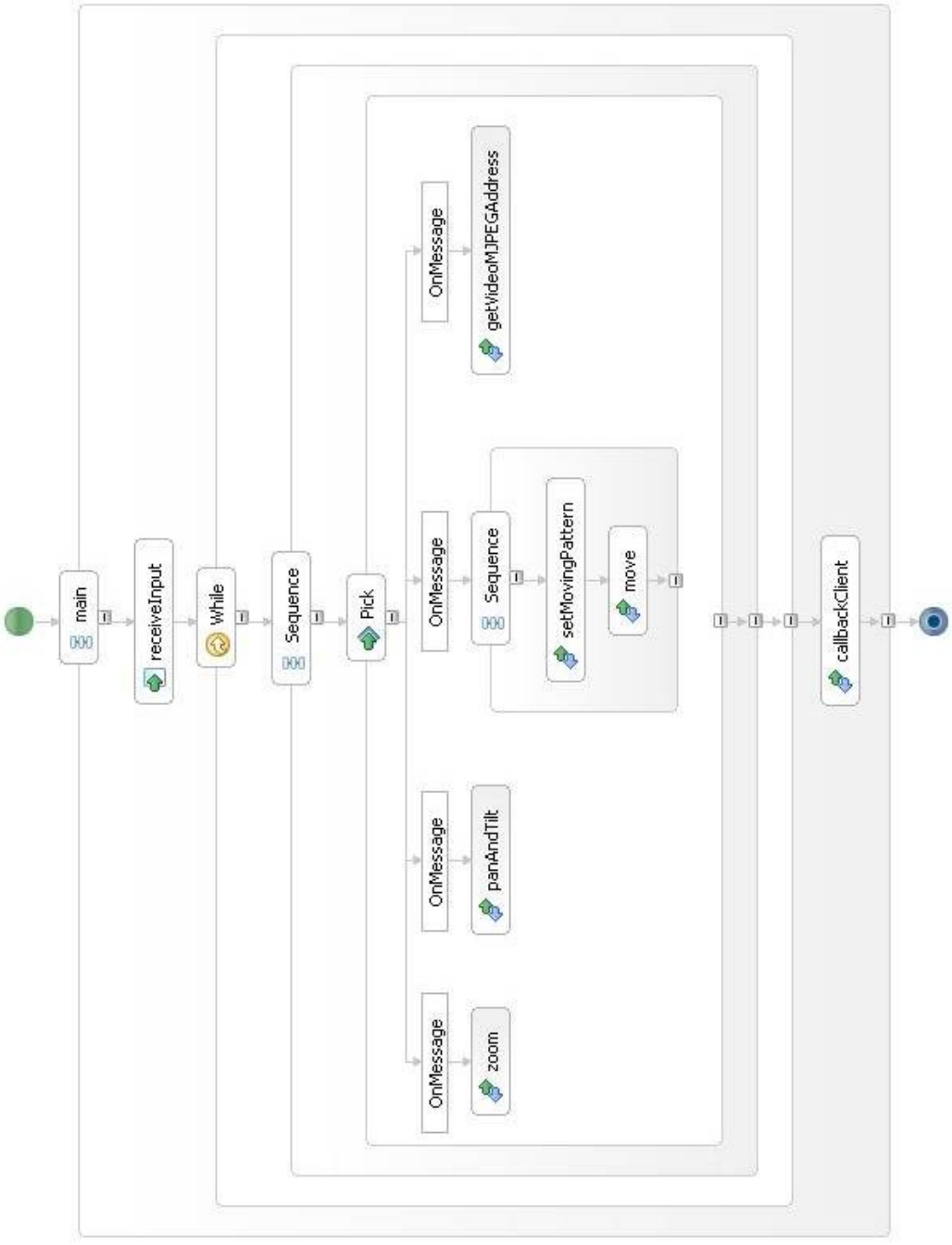

Figure 15: UGV Behaviour

3.3.4 Technical specifications

	Robot WifibOt Lab
	Motor sensor : 2 hall effect coders 2048 tics /wheel turn Battery level
	Speed control : 2 x PID DSPIC 30f2010
	Motors : 4x 12v motors 50:1 8.87Kg/cm 150 rpm
	Dimensions: L : 30 cm W : 35 cm H : 15 cm W : 3.8Kg
	Batteries: 9.6V NiMh 10000 mAH Charger 12V/220V
	Control bus I2C /USB / RS232
	Distant Protocol : Sockets TCP/UDP via WIFI or RJ45
	CPU : AMD Geode LX800 processor 1G Ram / 4G CF 4 x USB 2.0 4 x RS232/485
	Display type CRT & 24-bit TFT LCD
	Sensors : 2 Infrared 1 web cam Pan &Tilt
	Logiciels: C++ control API 2 HMI Embedded Web Server

Figure 16: WifiBoT lab Technical details

3.4 NS 3.1 - Camera Fixed

This NS is a RTSP video stream source, supporting zoom operation. It uses WS-discovery as Discovery Protocol.

3.4.1 Interfaces

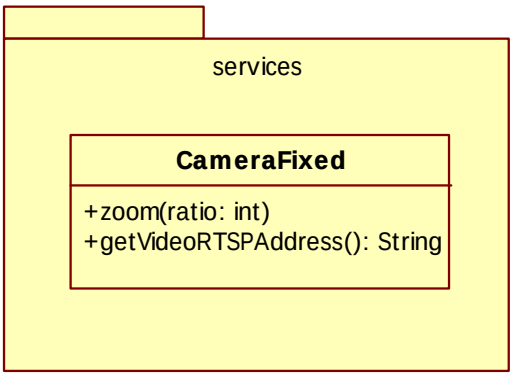

Figure 17: Camera fixed interface

zoom*(int ratio)* - sends a zoom command to the camera. *ratio* defines the relative zoom value

getVideoRTSPAddress*()* - returns the address of the video RTSP flux of the camera.

3.4.2 Affordance

This camera has the functional concept "*FixedVideoSource*" that is a sub class of "*VideoSource*". A client asking for a "*VideoSource*" will then be able to connect to this camera.

```
<?xml version="1.0" encoding="UTF-8"?>
<Affordances name="CameraFixed">
        <Affordance name="CameraFixedAff" kind="provided">

        <FunctionalConcept>http://www.connect.com/ontology/media#FixedVideoSource</FunctionalCon
cept>
                <Inputs>
                        <Input>http://www.connect.com/ontology/media#Void</Input>
                </Inputs>
                <Outputs>
                        <Output>http://www.connect.com/ontology/media#Void</Output>
                </Outputs>
        </Affordance>
</Affordances>
```

3.4.3 Behaviour

This fixed camera is relatively simple. It can respond to any operation "*zoom*" or "*getVideoRTSPAddress*".

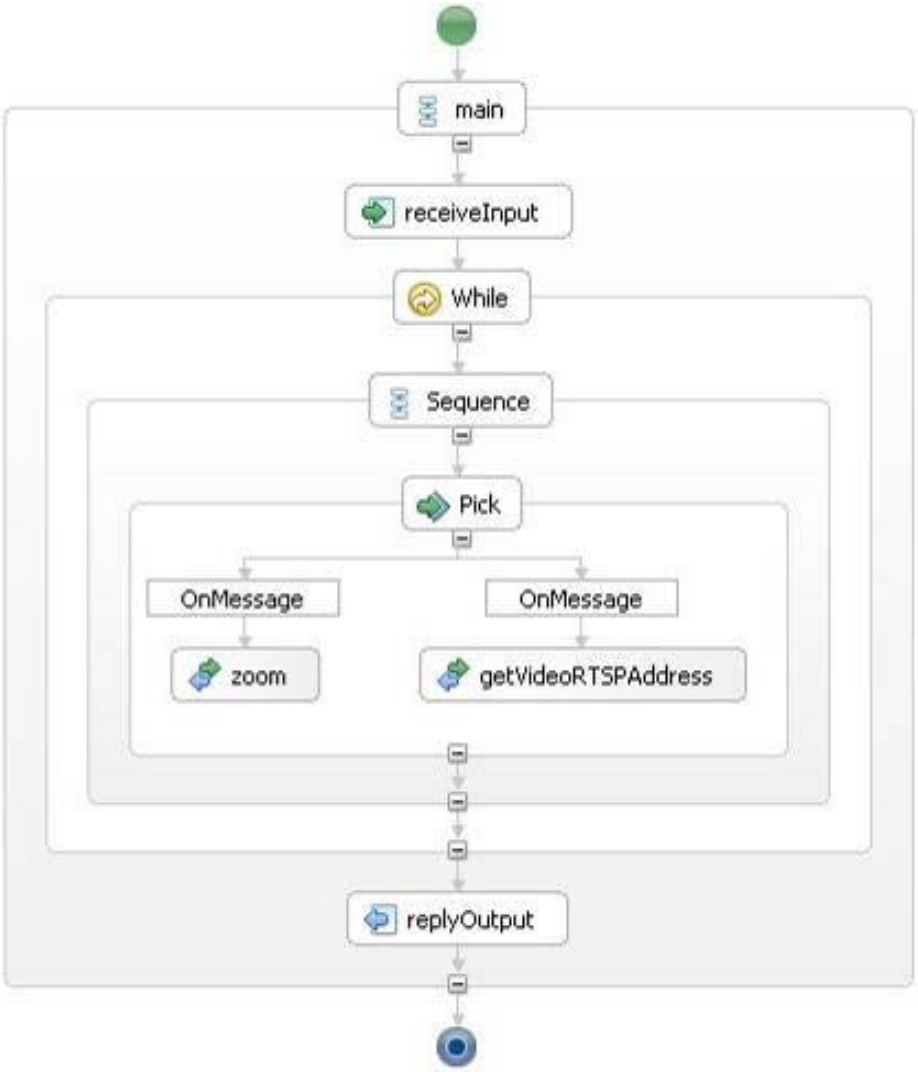

Figure 18: Fixed camera Behaviour

3.4.4 Technical specifications

	ELVIR MF
	- Spectral band: 8-12 µm - Resolution: 384 x 288 - Field of View: 6.9° x 5.1° (F = 80 mm) - - Digital magnetic compass - Operating time: 5 h ; NETD: 50 mK - Dimensions: 270 x 175 x 95 mm

Figure 19: ELVIR MF technical details

	**Axis 211**
	Image: Sensor Image Sensor 1⁄4” progressive scan CCD Lens: 3.0 - 80 mm, F1.2, DC-iris, CS mount Angle of view: 27° - 67° horizontal
	Video compression - Motion JPEG - MPEG-4 Part 2 Resolutions 16 resolutions from 640 x 480 to 160 x 120 via API, 5 selections via configuration web page. Frame rate (NTSC/PAL) - Motion JPEG:Up to 30 fps in all resolutions - MPEG-4: Up to 25 fps at 640x480Up to 30 fps at 480x360 or lower
	Dimensions and weight excl. power supply: - 38 x 95 x 178 mm (1.5” x 3.7” x 7.0”) - 250 g (0.55 lb) excl. power supply
	Open API for software integration including AXIS VAPIX API

Figure 20: Axis 211 technical details

3.5 NS 3.2 - Camera Mobile Main

This NS is a MJPEG video stream source that offers swivelling capabilities, thus supporting pan, tilt and zoom operations. It uses WS-Discovery as Discovery Protocol.

3.5.1 Interfaces

Figure 20: Camera Mobile Main interface

getVideoMJPEGAddress() - returns the address of the video MJPEG flux of the camera

panAndTilt*(float pan, float tilt)* - sends a pan/tilt command to the camera. *pan*: defines the relative pan value. *tilt*: defines the relative tilt value

3.5.2 Affordance

This camera has the functional concept "*MobileVideoSource*" that is a sub class of "*VideoSource*". A client asking for a "*VideoSource*" will then be able to connect to this camera.

```
<?xml version="1.0" encoding="UTF-8"?>
<Affordances name="CameraMain">
	<Affordance name="CameraMainAff" kind="provided">

	<FunctionalConcept>http://www.connect.com/ontology/media#MobileVideoSource</FunctionalCo
ncept>
		<Inputs>
			<Input>http://www.connect.com/ontology/media#Void</Input>
		</Inputs>
		<Outputs>
			<Output>http://www.connect.com/ontology/media#Void</Output>
		</Outputs>
	</Affordance>
</Affordances>
```

3.5.3 Behaviour

This mobile camera offers three operations that are totally independents: "*Zoom*", "*PanAndTilt*" and "*getVideoMJPEGAddress*".

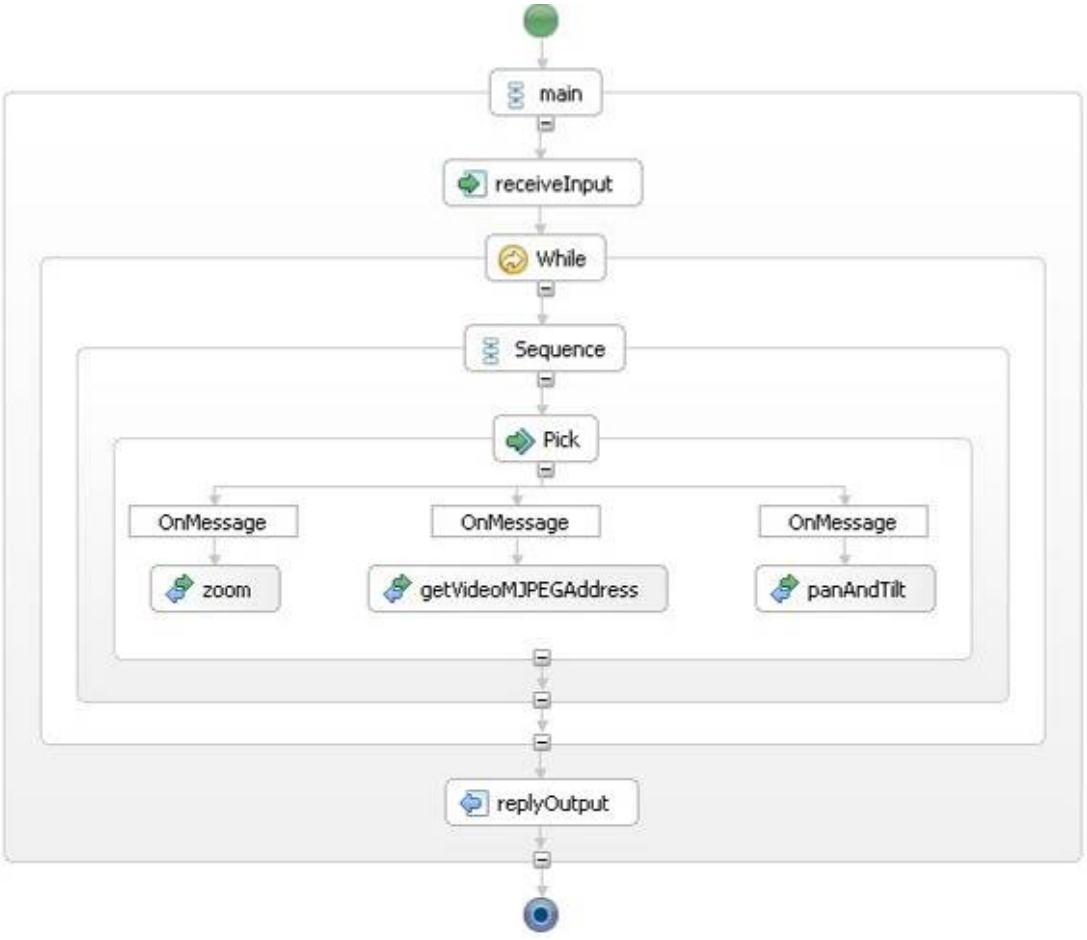

Figure 21: Mobile Main camera Behaviour

3.5.4 Technical specifications

	**Axis 213 PTZ**
	Image: Sensor Image Sensor 1⁄4” Interlaced CCD Lens: 3.5 - 91 mm, F1.6 – F4.0, motorized zoom lens, auto focus Angle of view: 1.7° - 47° horizontal Zoom: 26x optical, 12x digital
	Pan Range: ± 170° Tilt Range: -10 to 90° Max speed Pan: 1 - 90°/sec Max speed Tilt: 1 - 70°/sec
	Video compression - Motion JPEG - MPEG-4 Part 2 Resolutions 4CIF, 2CIFExp, 2CIF, CIF, QCIF - min 160x120 (NTSC) 176x144 (PAL) - max 704x480 (NTSC) 704x576 (PAL) Frame rate (NTSC/PAL) - Motion JPEG: Up to 30/25 fps at 4 CIF - MPEG-4: Up to 21/17 fps at 4CIF/2CIFExp
	Dimensions and weight excl. power supply: - 130 x 104 x 130 mm - 700 g
	Open API for software integration including AXIS VAPIX API

Figure 22: Axis 213 PTZ technical details

The AXIS 213 PTZ is a network camera that can be remotely orientated. The embedded zoom is also powerful enough to provide good surveillance capabilities. The camera contains:

- A video server delivering video streams of various types: Motion-JPEG or MPEG4 and over various protocols such as: RTP, RTSP, UDP.
- Some image processing algorithms such as motion detection.

The camera has to be powered by a 12V direct current and consumes approximately 500 mA. The power consumption is approximately 6W.

3.6 NS 3.3 - Camera Mobile Patrol

This NS offers MJPEG recorded video files, specifying recording time; it also supports a triggered patrol mode (on/off). It will use CDP as Discovery Protocol.

3.6.1 Interfaces

Figure 23: Camera Mobile Patrol interface

getTimedRecordedVideoLocalization*(int seconds)* - returns the absolute path address of the recorded video file of the camera. The length of the video is timed in seconds by the parameter *'seconds'*.

startPatrol*()* - orders the camera to start its patrol course

stopPatrol - orders the camera to stop its patrol course

3.6.2 Affordance

This camera has the functional concept "*PatrolVideoSource*" that is a sub class of "*VideoSource*". A client asking for a "*VideoSource*" will then be able to connect to this camera.

```
<?xml version="1.0" encoding="UTF-8"?>
<Affordances name="CameraPatrol">
	<Affordance name="CameraPatrolAff" kind="provided">

	<FunctionalConcept>http://www.connect.com/ontology/media#PatrolVideoSource</FunctionalCo
ncept>
		<Inputs>
			<Input>http://www.connect.com/ontology/media#Void</Input>
		</Inputs>
		<Outputs>
			<Output>http://www.connect.com/ontology/media#Void</Output>
		</Outputs>
	</Affordance>
</Affordances>
```

3.6.3 Behaviour

This camera has the ability to patrol following a predefined sequence of movements. Specifically, the operation "*startPatrol*" must be called before any access to operations "*zoom*" and "*getTimeRecordedVideoLocalization*". Afterwards, the operation "*stopPatrol*" can be called.

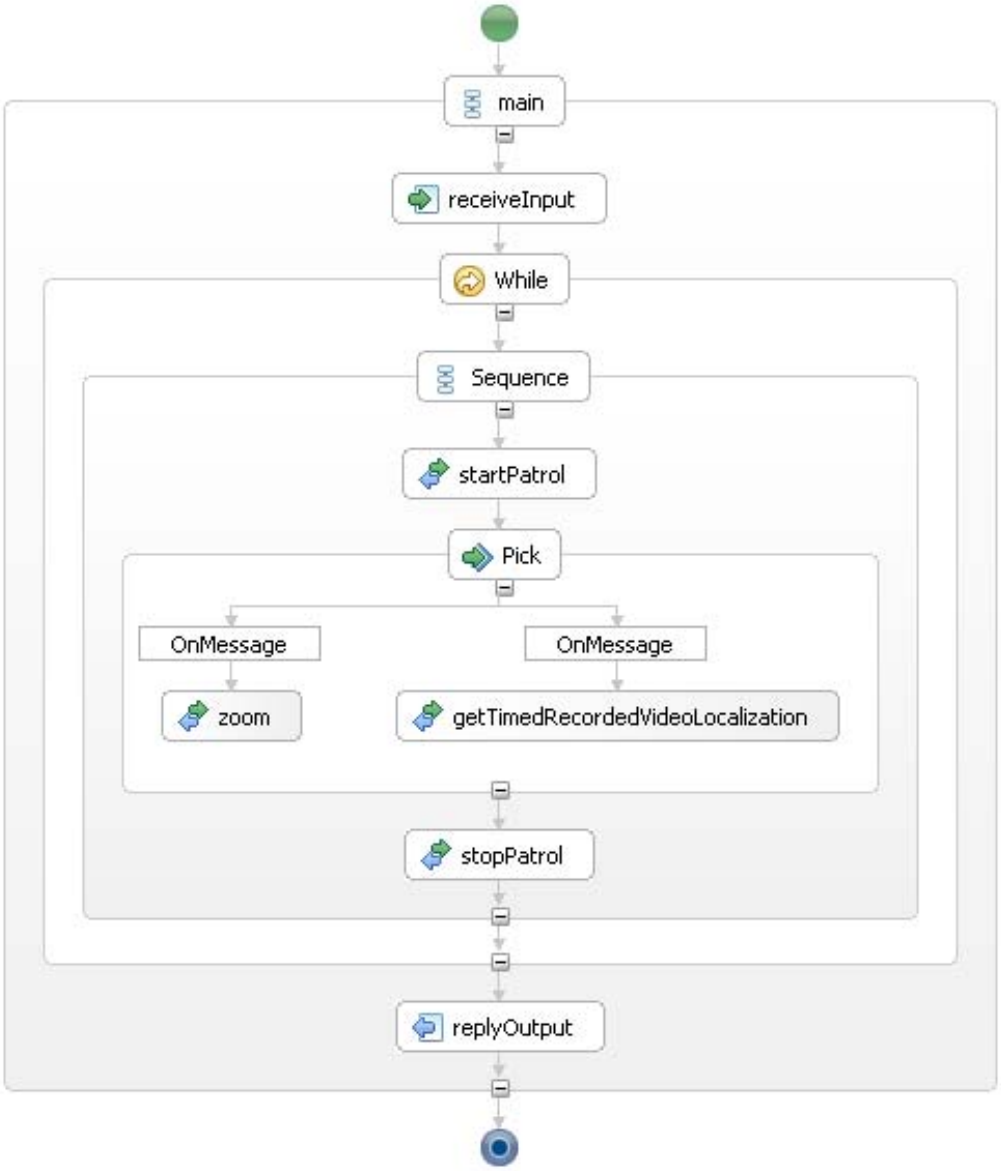

Figure 24: Mobile patrol camera Behaviour

3.6.4 Technical specifications

	**Axis 214 PTZ**
	Image: Sensor Image Sensor 1⁄4" HAD CCD Lens: 4.1 - 73.8 mm, F1.3 – F3.0, motorized zoom lens, auto focus Angle of view: 2.7° - 48° horizontal Zoom: 26x optical, 12x digital
	Pan Range: ± 170° Tilt Range: -30 to 90° Max speed Pan: 1 - 100°/sec Max speed Tilt: 1 - 90°/sec
	Video compression - Motion JPEG - MPEG-4 Part 2 Resolutions 4CIF, 2CIFExp, 2CIF, CIF, QCIF - min 160x120 (NTSC) 176x144 (PAL) - max 704x480 (NTSC) 704x576 (PAL) Frame rate (NTSC/PAL) - Motion JPEG: Up to 30/25 fps at 4 CIF - MPEG-4: Up to 25/21 fps at 4CIF/2CIFExp
	Dimensions and weight excl. power supply: - 165 x 157 x 151 mm - 1110 g
	Open API for software integration including AXIS VAPIX API

The AXIS 214 PTZ is a PTZ networked camera that has similar features as the PTZ camera introduced in Section 3.5.4.

The camera has to be powered by a 12V direct current and the power consumption is approximately 14W.

3.7 NS4 - C2 GIS

3.7.1 Interfaces

Unlike the others NS, C2 GIS acts only as a set of (GUI) clients, and more precisely the following:

- *CDP Client* displays the NSs discovered using the Discovery enabler. It is displayed as a tree map view (Resources panel in ***:*** *C2 GIS Display* - see Figure 27) allowing to select a resource, to connect to it or use this resource. And, if this resource is geo-referenced and already displayed as an icon on the map (see Figure 26 for icons), to highlight the icon with a bounding box.
- *Camera [Fixed, Mobile Main and Mobile Patrol, UAV Camera] client* allows to select a video resource and display its video stream, and, if the resource supports it, to send it control command (Live stream panel in ***:*** *C2 GIS Display*)
- *Position System Client* receives position data of a geo-referenced resource and displays its icon representation on the map according to the next table.

Icon	Resource Type	Comments
	Video source	Any kind camera , mobile or fixed
	Weather station	A full label appears on mouse over
	UAV	-
	UGV	-
	Fire Progression direction	Estimated direction for the evolution of a fire area
	Fire Area	Delimited fire area
	Fire point	Active fire point
	Fire Fighter	Can either represent 1 fighter or a team , depending on label
	Fire truck	-

Figure 25: C2 GIS Icon table

The C2 GIS NS also allows to send control commands to controllable resources using a generic control panel supporting basic moves [right| left| forward | backward |up | down] and the identified patrol patterns [Circle| Square | Diamond].

Figure 26: C2 GIS Display

3.7.2 Affordance

C2 GIS needs a lot of different type of services. It displays videos, it controls vehicles and cameras. It needs weather data and position of individuals. This is reflected in its affordances: "*VideoSource*" for any kind of video, "*WeatherInfo*" for weather data, "*MobileVideoSource*" for video whose recorder is movable, "*Vehicle*" for movable transport, and "*PositionSource*" for positions feeders.

```
<?xml version="1.0" encoding="UTF-8"?>
<Affordances name="C2">
	<Affordance name="video" kind="required">

	<FunctionalConcept>http://www.connect.com/ontology/media#VideoSource</FunctionalConcept>
		<Inputs>
			<Input>http://www.connect.com/ontology/media#Void</Input>
		</Inputs>
		<Outputs>
			<Output>http://www.connect.com/ontology/media#Void</Output>
		</Outputs>
	</Affordance>
	<Affordance name="weather" kind="required">

	<FunctionalConcept>http://www.connect.com/ontology/media#WeatherInfo</FunctionalConcept>
		<Inputs>
			<Input>http://www.connect.com/ontology/media#Void</Input>
		</Inputs>
		<Outputs>
			<Output>http://www.connect.com/ontology/media#Void</Output>
		</Outputs>
	</Affordance>
	<Affordance name="vehiclevideo" kind="required">

	<FunctionalConcept>http://www.connect.com/ontology/media#MobileVideoSource</FunctionalCo
ncept>
		<Inputs>
			<Input>http://www.connect.com/ontology/media#Void</Input>
		</Inputs>
		<Outputs>
			<Output>http://www.connect.com/ontology/media#Void</Output>
		</Outputs>
	</Affordance>
	<Affordance name="vehicle" kind="required">

	<FunctionalConcept>http://www.connect.com/ontology/media#Vehicle</FunctionalConcept>
		<Inputs>
			<Input>http://www.connect.com/ontology/media#Void</Input>
		</Inputs>
		<Outputs>
			<Output>http://www.connect.com/ontology/media#Void</Output>
		</Outputs>
	</Affordance>
	<Affordance name="position" kind="required">
```

```
        <FunctionalConcept>http://www.connect.com/ontology/media#PositionSource</FunctionalConce
pt>
                <Inputs>
                        <Input>http://www.connect.com/ontology/media#Void</Input>
                </Inputs>
                <Outputs>
                        <Output>http://www.connect.com/ontology/media#Void</Output>
                </Outputs>
        </Affordance>
</Affordances>
```

3.7.3 Behaviour

The first action launched by the C2 GIS is the operation "*subscribePositioning*" that connects to positioning systems. Then the C2 GIS can independently access to three different phases: (i) the control phase of video sources (operations "*panAndTilt*", "*zoom*", "*getVideo*"); (ii) the meteo phase ("*getWeather*"); and (iii) the vehicle control phase ("*moveVehicle*").

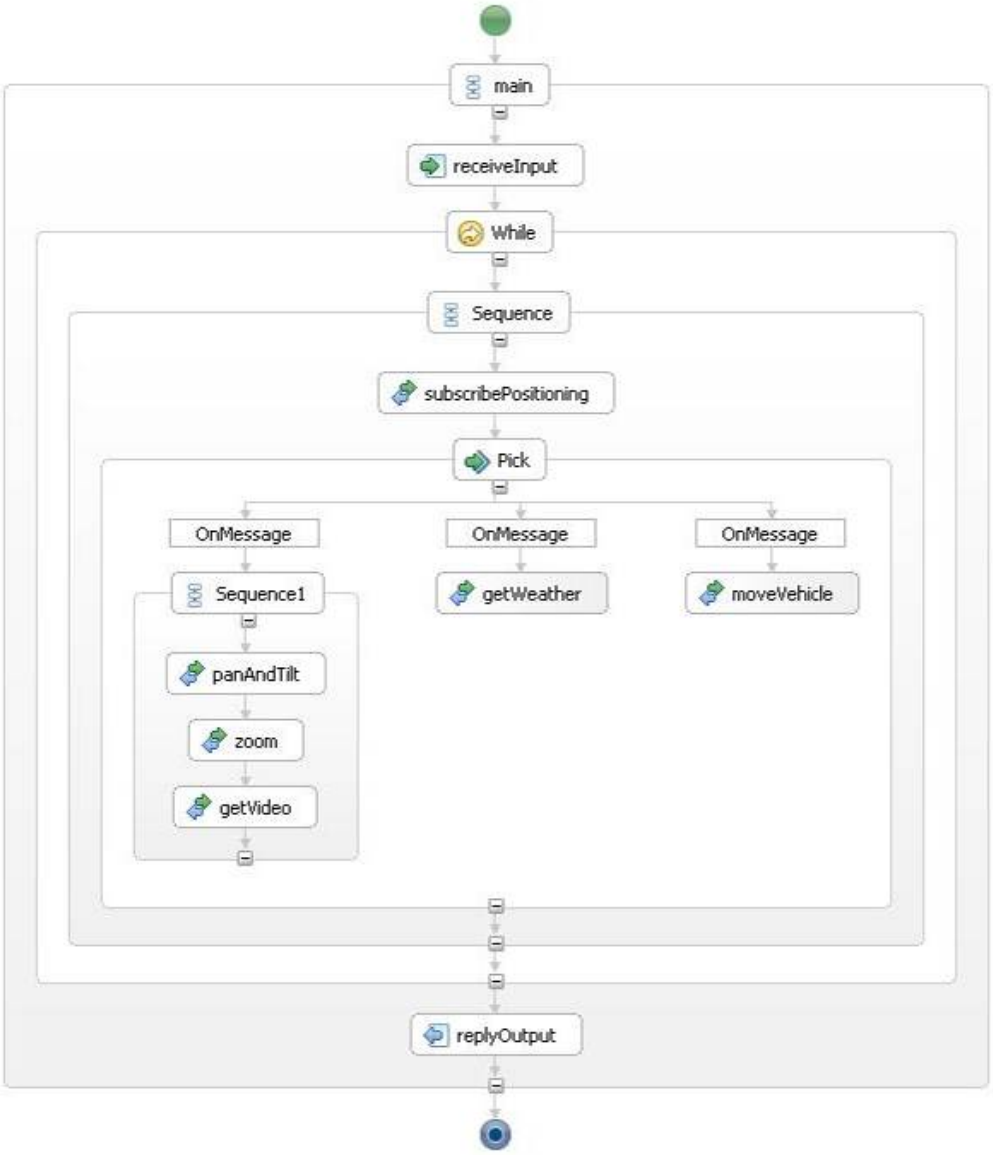

Figure 27: C2 GIS Behaviour

3.7.4 Technical specifications

Cartographic display and GUI should be realized using GWT.

3.8 NS 5 - Mobile Weather Station

This system provides local weather information through a shared data space exchange implemented with Lime. It uses CDP as Discovery Protocol.

3.8.1 Interfaces

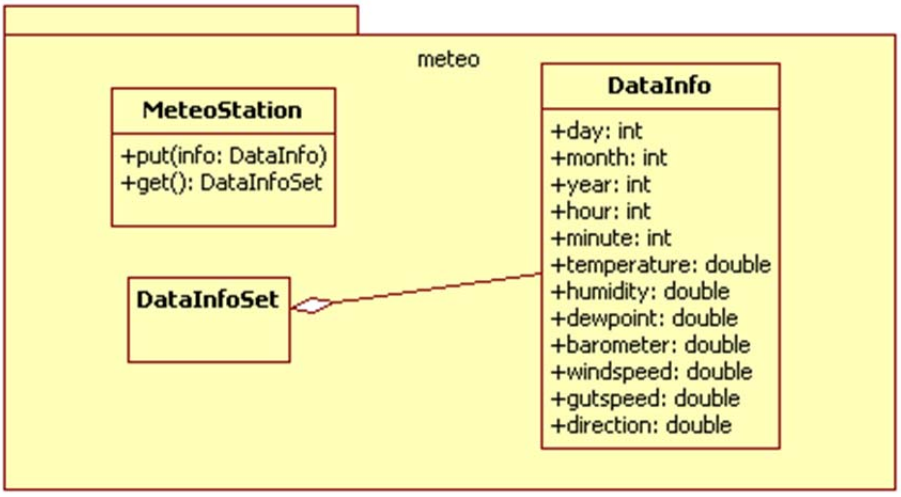

Figure 28: Weather Station interface

put*(DataInfo info)* - records the information on the station. *info* : data gathered by the station

get() - returns the set of data information recorded by the meteo station

3.8.2 Affordance

The weather station has the functional concept "*WeatherStory*" that is a sub class of "*WeatherInfo*". A client asking for a "*WeatherInfo*" will then be able to connect to this station.

```
<?xml version="1.0" encoding="UTF-8"?>
<Affordances name="WeatherStation">
	<Affordance name="WeatherStationAff" kind="provided">

	<FunctionalConcept>http://www.connect.com/ontology/media#WeatherStory</FunctionalConcept
>
		<Inputs>
			<Input>http://www.connect.com/ontology/media#Void</Input>
		</Inputs>
		<Outputs>
			<Output>http://www.connect.com/ontology/media#Void</Output>
		</Outputs>
	</Affordance>
</Affordances>
```

3.8.3 Behaviour

Client using this station can do two things independently: one is to "*put*" weather information. Second is to "*get*" weather information.

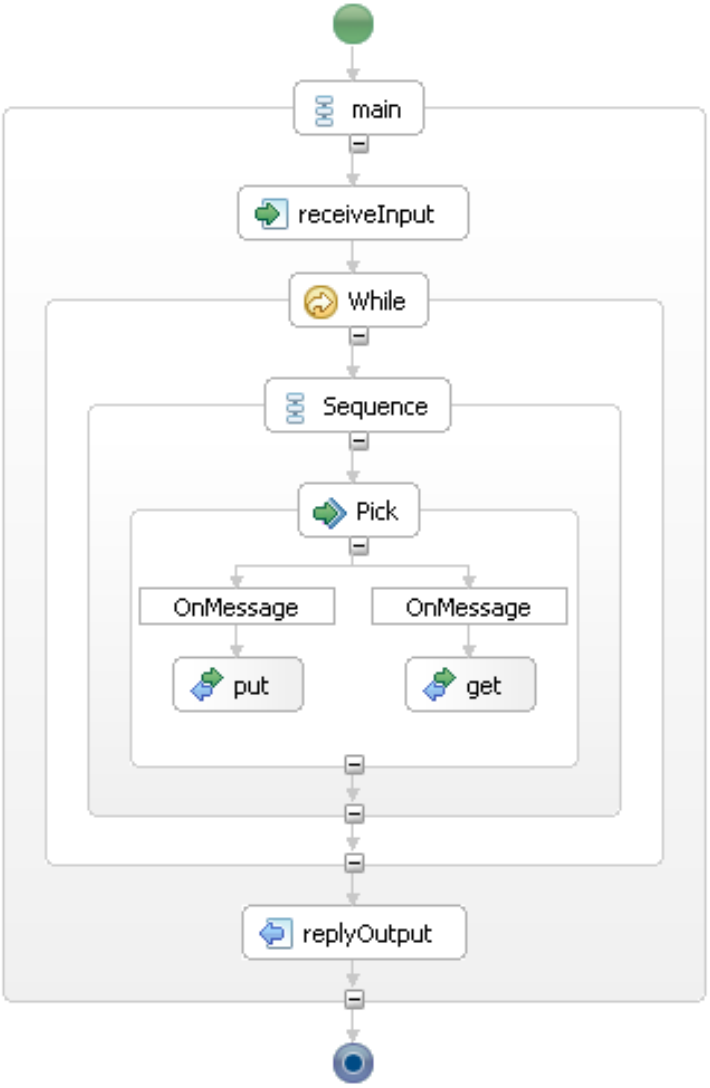

Figure 29: Weather station Behaviour

3.8.4 Technical specifications

The foreseen weather station is a Lacross Technology WS 2 550 composed of:

- A control panel that displays the weather information on a LCD display
- A remote multi sensors set wirelessly connected to the station

The figure below shows the display provided by the weather station WS 2 550.

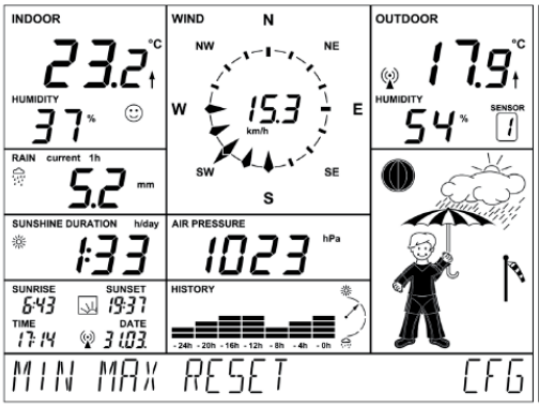

Figure 30: control panel of the WS 2 550

The control panel of the weather station is hosting a micro controller and may be connected to a computer thanks to a USB connection.

The figure below shows the multi sensors set of the weather station WS 2 550.

Figure 31: sensors set of the WS 2 550

The sensors are deployed at the top of a mast and send the following data through a wireless connection to the control panel:

- Temperature
- Hygrometry
- Wind direction and speed
- Pluviometry
- Sunny timing

The data gathered periodically by the weather station software hosted in the computer are inserted in a dedicated web service. In the context of CONNECT, we exploit the temperature, hygrometry and the wind characteristics.

3.9 NS 6 - Weather Service

This service provides weather report on a given location using a SOAP Web service. It uses CDP as Discovery Protocol.

3.9.1 Interfaces

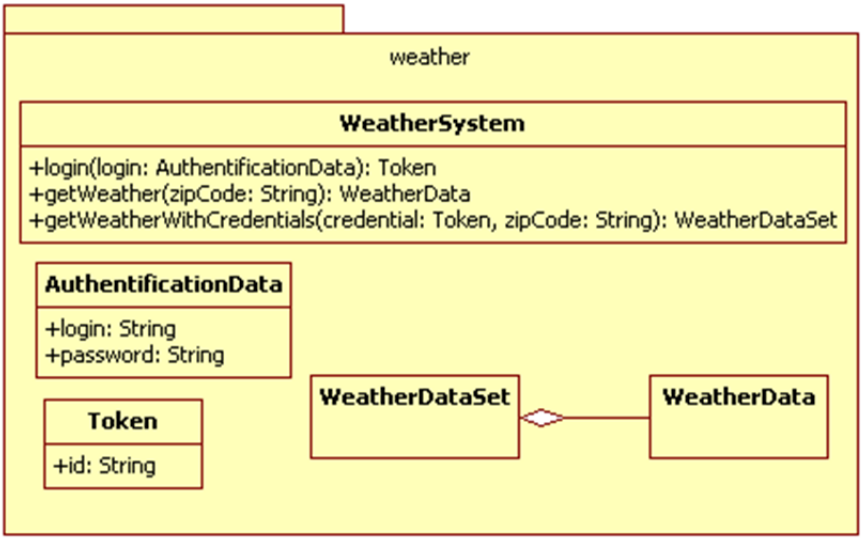

Figure 32: Weather Service interface

login*(AuthentificationData login)* - logs the user into the system using his login/password. It grants the user with a token when login succeeds. *login* : user information

getWeather*(String zipCode)* - returns the current weather data of the given region. *zipCode* : location area from which the weather will be sought.

getWeatherWithCredentials*(AuthentificationData login, String zipCode)* - returns all the current and future weather data of the given region.

3.9.2 Affordance

The weather service has the functional concept "*WeatherWithForcast*" that is a sub class of "*WeatherInfo*". A client asking for a "*WeatherInfo*" will then be able to connect to this service.

```
<?xml version="1.0" encoding="UTF-8"?>
<Affordances name="WeatherService">
	<Affordance name="WeatherServiceAff" kind="provided">

	<FunctionalConcept>http://www.connect.com/ontology/media#WeatherWithForecast</Functional
Concept>
		<Inputs>
			<Input>http://www.connect.com/ontology/media#Void</Input>
		</Inputs>
		<Outputs>
			<Output>http://www.connect.com/ontology/media#Void</Output>
		</Outputs>
	</Affordance>
</Affordances>
```

3.9.3 Behaviour

This weather service has the ability to give the current weather information using the "*getWeather*" operation or the weather forecast. To get a weather forcast the sequence of operation "*login*" and "*getWeatherWithCredentials*" must be called.

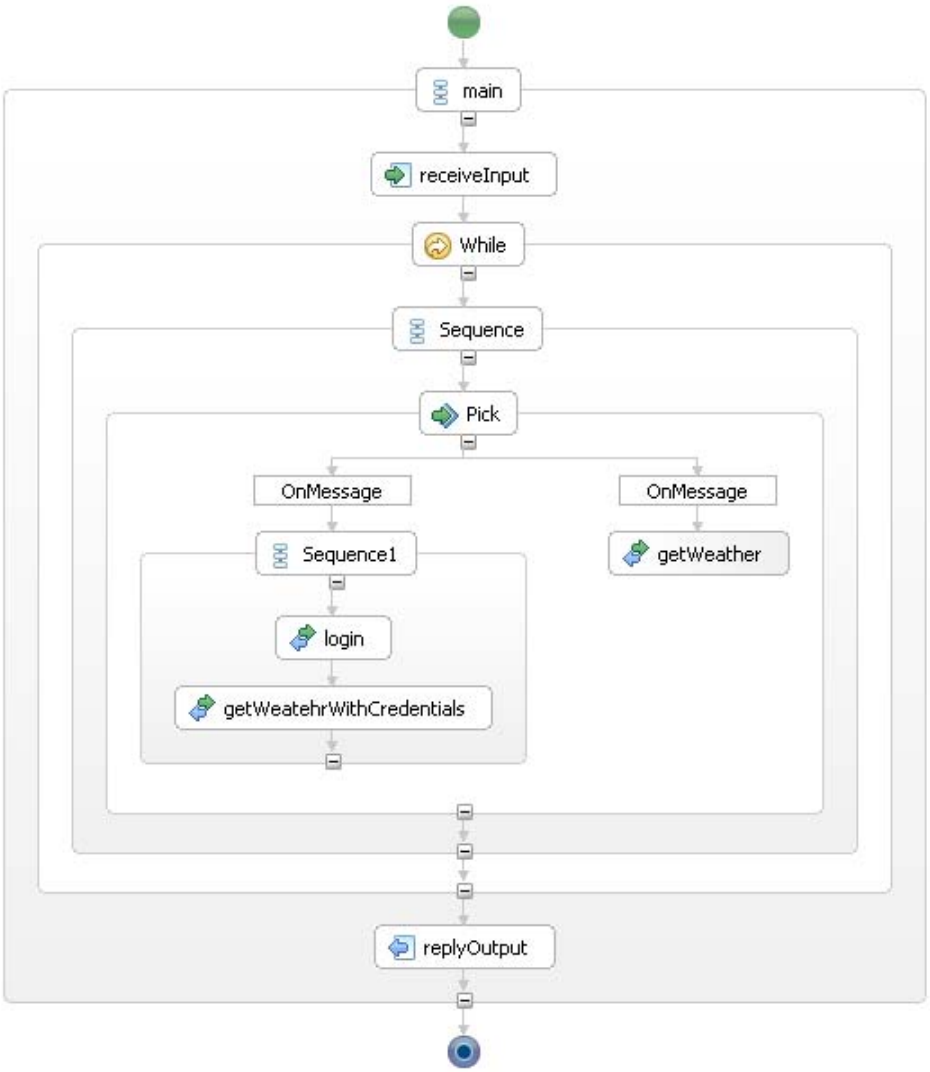

Figure 33: Weather service Behaviour

3.9.4 Technical specifications

This service should expose standard Web service interfaces (SOAP 1.1 or 1.2)

3.10 NS 7.1 - Positioning System – Country A

This system provides information on location of Country A resources, using a Publish / Subscribe interaction protocol that is realized with OMG DDS. It uses CDP as Discovery Protocol.

3.10.1 Interfaces

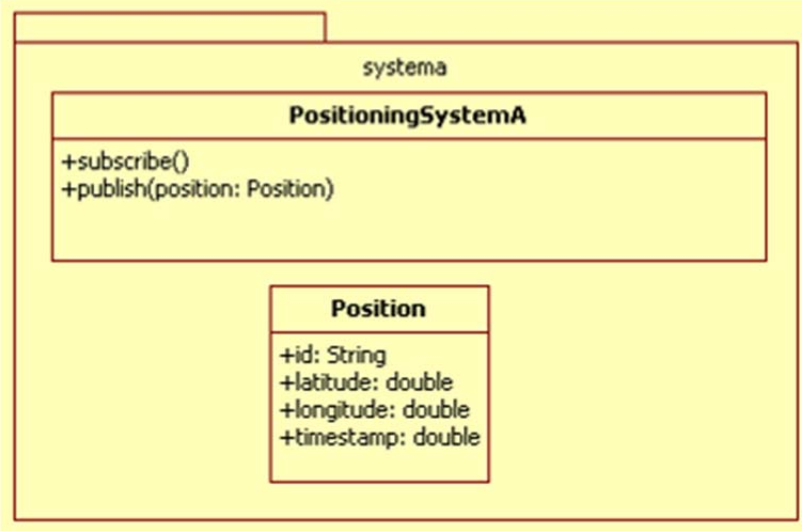

Figure 34: Positioning system interface

subscribe() - subscribes for position information.

publish*(String position)* - publishes a position. *Position*: the data for a position in expected format

3.10.2 Affordance

This service has the functional concept "*PositioningSourceDDS*" that is a sub class of "*PositioningSource*". A client asking for a "*PositioningSource*" will then be able to connect to this service.

```
<?xml version="1.0" encoding="UTF-8"?>
<Affordances name="PositioningDDSSystem">
	<Affordance name="PositioningDDSSystemAff" kind="provided">

	<FunctionalConcept>http://www.connect.com/ontology/media#PositioningSourceDDS</Functiona
lConcept>
		<Inputs>
			<Input>http://www.connect.com/ontology/media#Void</Input>
		</Inputs>
		<Outputs>
			<Output>http://www.connect.com/ontology/media#Void</Output>
		</Outputs>
	</Affordance>
</Affordances>
```

3.10.3 Behaviour

This service is relatively simple. It can respond to any operation "*subscribePositioning*" or "*publishPosition*".

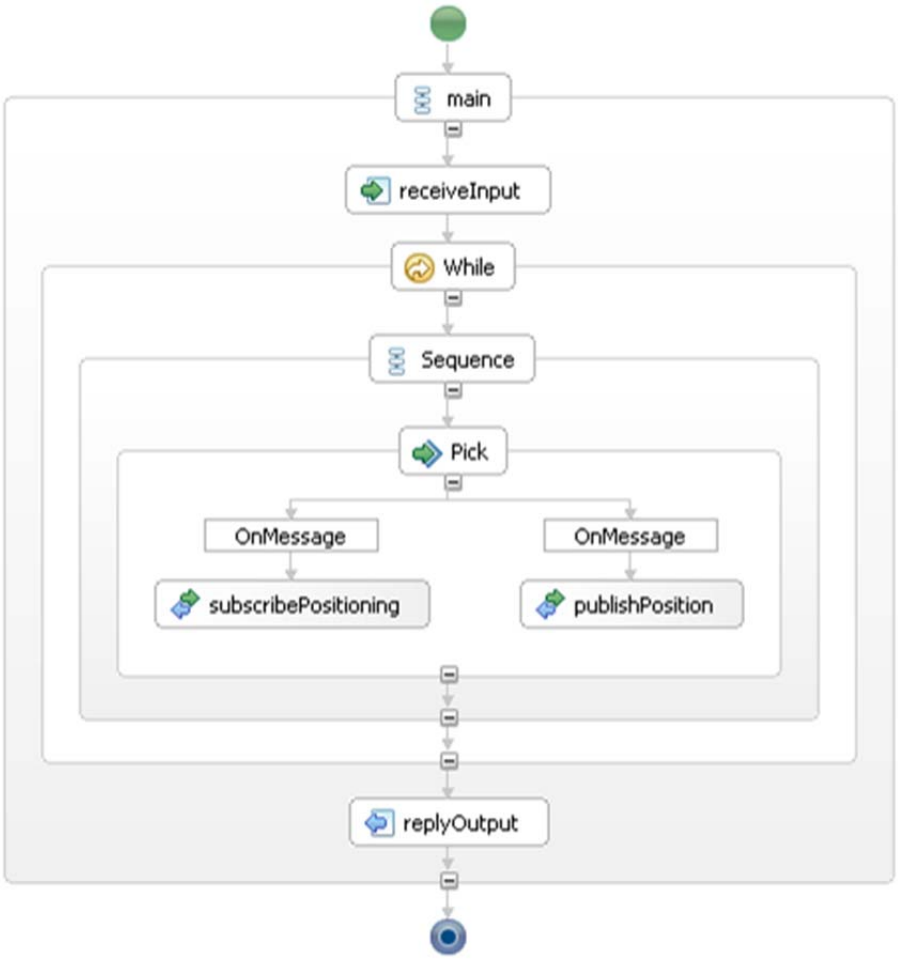

Figure 35: Positioning system A Behaviour

3.11 NS 7.2 - Positioning System – Country B

This system provides information on Country resource location using a Message exchange pattern. It uses CDP as Discovery Protocol.

3.11.1 Interfaces

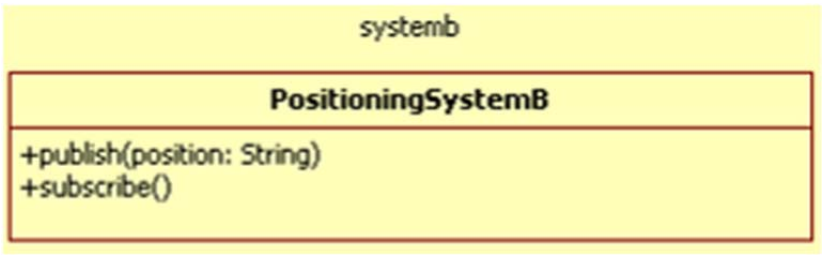

Figure 36: Positioning System B interface

subscribe() - subscribes to information.

publish*(String position)* - publishes a position to an identified topic. *position*: a string delimited value corresponding to the position following the format ID;LATITUDE;LONGITUDE;TIMESTAMP

3.11.2 Affordance

This service has the functional concept "*PositioningSourceJMS*" that is a sub class of "*PositioningSource*". A client asking for a "*PositioningSource*" will then be able to connect to this service.

```
<?xml version="1.0" encoding="UTF-8"?>
<Affordances name="PositioningJMSSystem">
	<Affordance name="PositioningJMSSystemAff" kind="provided">

	<FunctionalConcept>http://www.connect.com/ontology/media#PositioningSourceJMS</Functiona
lConcept>
		<Inputs>
			<Input>http://www.connect.com/ontology/media#Void</Input>
		</Inputs>
		<Outputs>
			<Output>http://www.connect.com/ontology/media#Void</Output>
		</Outputs>
	</Affordance>
</Affordances>
```

3.11.3 Behaviour

This service is relatively simple. It can respond to any operation "*subscribePositioning*" or "*publishPosition*".

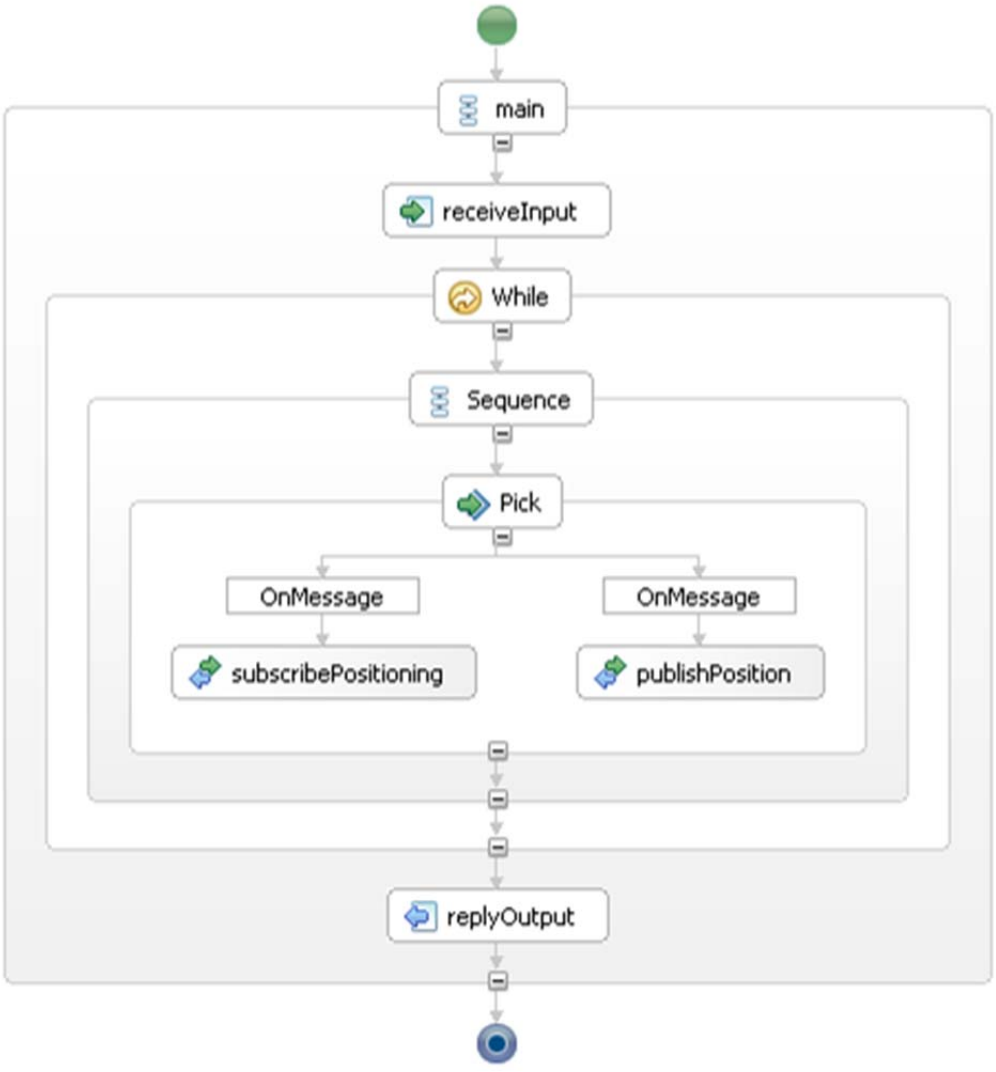

Figure 37: Positioning system B Behaviour

3.12 NS 8 - Firefighter Live Multimedia Communication

The scenario will also contain a live videoconference system for firefighters using which the C2 is able to initiate multimedia communications with squad leaders requiring audio and/or video streams setup and adaptation across heterogeneous mobile devices. Supported interaction models will include Facetime-like and push2talk (walkie-talkie) communication.

This NS is in the very early stages of development, and more details will be provided in D6.4.

4 DEMONSTRATOR OVERVIEW

This chapter introduces the demonstrator that we are developing based on the GMES scenario related to Joint Forest-Fire operation that was discussed in Chapter 2. The purpose of the demonstrator is to enable the assessment of the CONNECT solutions with respect to effectively enabling on-the-fly interoperability across heterogeneous networked systems.

We first review the use of CONNECT enablers in the realization of the Joint-Forest operation scenario. Then, we outline the CONNECTors to be synthesized, followed by the definition of the milestones associated with the implementation of the demonstrator.

4.1 Joint Forest-Fire Operation using CONNECT

Based on the scenario introduced in chapter 2, this section discusses the CONNECT-specific objectives associated with the realization of some of the steps that are the ones with which we will experiment CONNECT enablers

For each objective, we specify: the Enablers involved, the requirements addressed by CONNECT, and challenges to be tackled.

Step 1.2 - Deployment of sensors and operational forces

Objective 1: Discovery of existing resources

Enablers Involved: DISCOVERY, LEARNING

Need: To access to a list of available resources: the UGV, cameras, and positioning system

Challenges: Multi-protocol discovery and learning to complete partial descriptions

Objective 2: Usage of identified matching resources

Enablers Involved: DISCOVERY, SYNTHESIS, INVOCATION, DEPLOYMENT

Need: To connect systems needing to interoperate together, such as the C2 and Cameras, C2 and Positioning Systems

Challenges: Synthesis and deployment of appropriate CONNECTors between matching systems with heterogeneous interfaces and protocols

Objective 3: Usage of various exchange patterns

Enablers Involved: SYNTHESIS, INVOCATION and DEPLOYMENT

Need: All pairs at this stage do not use only RPC.

Challenges: Synthesis and deployment of appropriate CONNECTors between matching systems with heterogeneous interfaces and protocols

Step 1.3 - Fire-fighting phase

Objective: System operation

Enablers Involved: MONITORING

Need: The CONNECTed system must perform as expected while running

Challenges: Adequate observation of relevant non-functional properties

Step 1.4 - Fire evolution phase and cross-country deployment

Objective: Expose secured resource

Enablers Involved: SYNTHESIS, INVOCATION, MONITORING, DEPLOYMENT SECURITY and TRUST.

Need: The resources of country B such as the UAV must be used securely.

Challenges: Instantiate a Connector between two NSs with one requesting to comply with a security policy

Step 2.1 - Discovery of new resources

Objective: Discovery of newly deployed resources

Enablers Involved: DISCOVERY and LEARNING.

Need: The resources from Country B (Weather Service, UAV and Positioning system B) should be discovered

Challenges: Multi-protocol discovery and learning to complete partial descriptions

Step 2.2 - Connection to Country B Position Service

Objective: Consume unexpected / unknown resources.

Enablers Involved: SYNTHESIS, INVOCATION and DEPLOYMENT.

Need: C2 must interact with the Positioning System B.

Challenges:

- Instantiate a CONNECTor between two NSs using same exchange pattern realized with different transport protocol.
- Instantiate a CONNECTor between two NSs using different data format.

Step 2.3 - Consume the Country B Weather service

Objective: Consume unexpected / unknown resources.

Enablers Involved: SYNTHESIS, INVOCATION, DEPLOYMENT and SECURITY.

Need: The weather service using RPC must interact with C2 using a shared space pattern.

Challenges: Instantiate a CONNECTor between two NSs using different transport protocols, data formats, and exchange patterns.

Step 2.4 - Fire-fighting phase

Objective 1: System operation.

Enablers Involved: MONITORING.

Need: The CONNECTed system must continue performing as expected.

Challenges: Adequate observation of relevant non-functional properties.

Objective 2: Use secured resources

Enablers Involved: SYNTHESIS, INVOCATION, MONITORING, DEPLOYMENT SECURITY and TRUST

Need: Modification of the UAV flight pattern requires authentication

Challenges: Instantiate a Connector between two NSs with one requesting to comply with a security policy

Live Multimedia Communications as part of the scenario steps

Objective: Video stream adaptation

Enablers Involved: SYNTHESIS, INVOCATION, MONITORING, DEPLOYMENT and DePer.

Need: Video streams between the cameras and consumers such as C2 or fire-fighters must be handled.

Challenges: Instantiate a Connector between two NSs for video streaming consumption that requires video format adaptation.

4.2 NSs and Connectors Map

The following table provides a summary of the interaction paradigms and protocols implemented by the various networked systems, which need to be mediated so as to enable the CONNECTion of the GMES-based NSs depicted in Figure 38.

Interaction paradigm	Protocol	NSs	Comments
RCP	SOAP	Weather Service, UAV	
	DPWS	Camera Mobile Main, Camera Fixed	For WS-Discovery
	HTTP	UGV, Camera Mobile Patrol	
Stream	RTSP	Camera Fixed, UAV Camera	
	MJPEG	Camera Mobile Main	
Pub/Sub	DDS	Positioning System A	Dedicated topic Position Position.idl file
	AMQP		Dropped
Message	JMS	Positioning System B	Dedicated topic Position Using text message
Shared Space	Lime	Weather Station	
	JavaSpace		Dropped

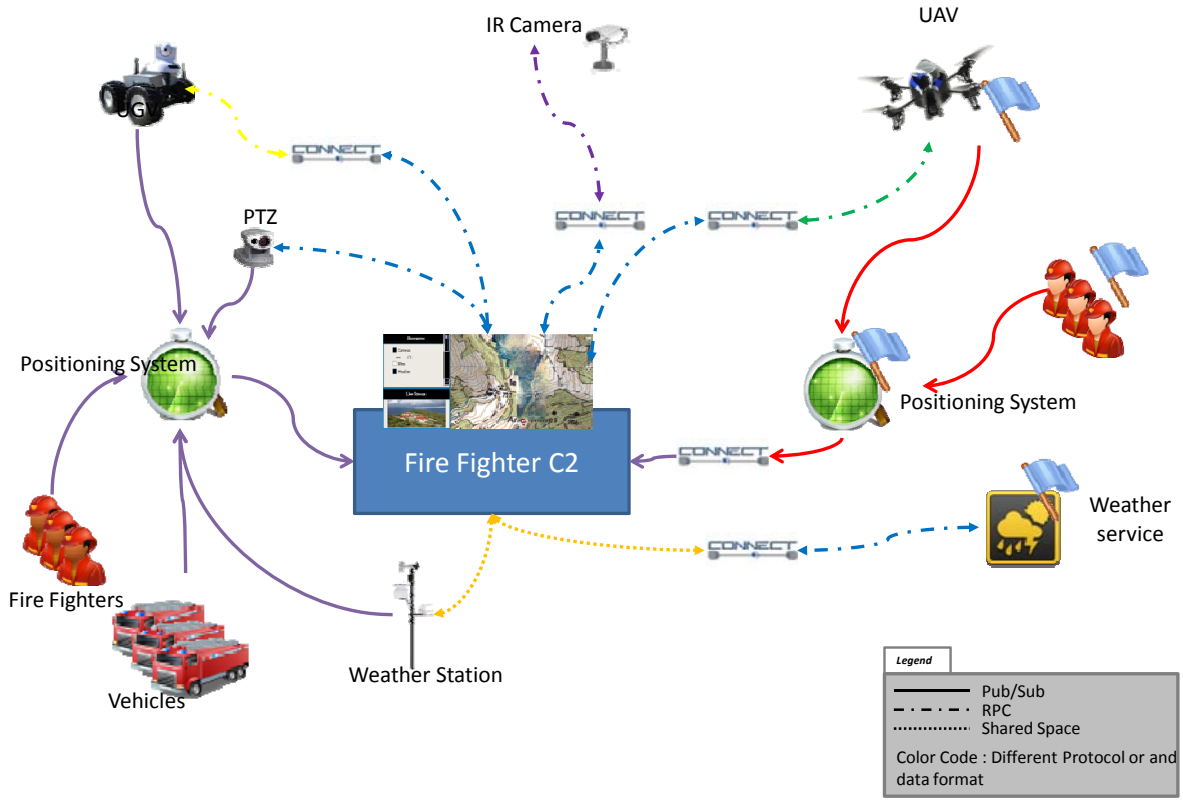

Figure 38: Demonstrator - NSs & CONNECTors Map

4.3 Milestones and Status

There are six identified milestones for the development of the GMES-based demonstrator (see Table 1): three for Enablers Integration (EI-x) and three Demonstrator versions (D-x).

ID	Date	Contents
EI-0	01/02/2012	1st version of the enablers integration
D-1	01/03/2012	1st version of demonstrator that should demonstrate Discovery and Learning
EI-1	01/06/2012	2nd version of the enablers integration
D-2	13/06/2012	2nd version of demonstrator that should include D-1 plus Interaction, Synthesis, Deployment and optionally Monitoring and Security.
EI-Final	01/10/2012	Final version of the enablers integration
D-Final	31/10/2012	Final version of the demonstrator that should include DePer & Trust demos

Table 1: Milestones

More details on each milestone contents are provided in **Table 2**, while Table 3 provides the current status of the different networked systems to be realized for the demonstrator, where TBS indicates that the work is “To Be Started” In the next period.

Source code of the prototypes are available on the CONNECT svn under the WP6/NS/Vx directory.

	EI-0	D-1	EI-1	D-2	EI-Final	D-3
Date	01/02/2012	01/03/2012	01/06/2012	13/06/2012	01/10/2012	31/10/2012
Enablers						
E1 Discovery						
E2 Learning						
E3 Synthesis						
E4 Deployment						
E5 Monitoring						
E6 Interaction						
E7 DePer						
E8 Security						
E9 Trust						
Networked Systems						
NS 1.1 UAV						
NS 1.2 UAV Camera						
NS 2 UGV						
NS 3.1 Camera Fixed						
NS 3.2 Camera Mobile Patrol						
NS 3.3 Camera Mobile Main						
NS 4 C2						
NS 5 Weather Station						
NS 6 Weather Service						
NS 7.1 Position System A						
NS 7.2 Position System B						
NS 8 Live Multimedia Communication						
LEGEND		Yes		Optional / Part		No

Table 2: Milestones contents

ID	Name	Interface	Affordance	Behavior	Implementation versio	EI-version	Comments
NS 1.1	UAV	Y	Y	Y	In progress	TBS	1 st version without enablers support available at Y3 demo
NS 1.2	UAV Camera	Y	Y	Y	In progress	TBS	1 st version without enablers support available at Y3 demo
Ns 2	UGV	Y	Y	Y	TBS	TBS	
	UGV Camera	Y	Y	Y	TBS	TBS	
NS 3.1	Camera Fixed	Y	Y	Y	1.0	TBS	1 st version without enablers support
NS 3.2	Camera Mobile Patrol	Y	Y	Y	1.0	TBS	1 st version without enablers support
NS 3.3	Camera Mobile Main	Y	Y	Y	1.0	TBS	1 st version without enablers support
NS 4	C2	Y	Y	Y	1.0	TBS	1 st version without enablers support
	Live Multimedia Communications C2	TBS	TBS	TBS	TBS	TBS	
	Weather Client	Y	Y	TBS	In progress	TBS	
	Position Client	Y	Y	Y	1.0	TBS	1 st version without enablers support
NS 5	Weather Station	Y	Y	Y	In progress	TBS	
NS 6	Weather Service	Y	Y	Y	1.0	TBS	1 st version without enablers support
NS 7.1	Position System A	Y	Y	Y	1.0	TBS	1 st version without enablers support
NS 7.2	Position System B	Y	Y	Y	1.0	TBS	1 st version without enablers support
NS 8	Live Multimedia Communications Server	TBS	TBS	TBS	TBS	TBS	
	Live Multimedia Communications Client	TBS	TBS	TBS	TBS	TBS	

Table 3: NSs Status

5 ASSESSMENT

While the previous chapters have detailed the GMES-based scenario and related demonstrator that we are developing to experiment with CONNECT solutions, this chapter discusses the methodology that we will apply to assess CONNECT results against CONNECT's ambitious objective of sustaining eternal interoperability.

We first outline the specific objectives associated with each CONNECT work package that, when combined, achieve the project's overall aim of eternal interoperability between heterogeneous networked software systems. Then, for each objective, we define the criteria of success and our methodology for qualitative evaluation based on the GMES scenarios and demonstrator provided by WP6. Preliminary assessment based on the project results at M36 is also sketched, while thorough assessment will be reported in the next, final period.

5.1 CONNECT Concepts and Challenges

As introduced in the project's Description of Work, CONNECT shall sustain future-proof dynamic networking among any digital systems, from the legacy to the yet-to-come, by dynamically generating mediators (a.k.a. *CONNECTORS*), thus bringing universal and eternal interoperability. CONNECTors are thus synthesized on the fly according to the behavioural semantics of application- down to middleware-layer protocols run by the interacting parties. Also, CONNECTors are dependable, unobtrusive, and evolvable to indeed meet the promise of eternality, while not compromising the quality of software applications. Last but not least, CONNECTors are concrete system entities, being implemented on the fly to enable actual interactions.

In order to support the above on-the-fly synthesis of CONNECTors, the following challenges, as described in the project's Description of Work, must be addressed:

1. **Modelling and reasoning about peer system functionalities**: A holistic and detailed model of networked systems is necessary in order to support accurate compatibility checks and subsequent synthesis and monitoring of a mediator between a compatible pair. In particular, establishing ontological relationships between systems and their functions is critical.
2. **Modelling and reasoning about connector behaviours**: A framework for specifying the behaviour of CONNECTors and elaborating such specifications to support runtime synthesis techniques is required.
3. **Runtime synthesis of CONNECTors:** From the abstract specifications of networked systems and CONNECTors, efficient techniques must be provided that synthesise concrete CONNECTors which can be directly deployed. Concretisation entails consideration of multiple layers of abstractions, from the application through middleware to the network layer.
4. **Learning systems behaviours:** Achieving a complete networked system description in practical cases where a complete description is not provided requires the use of machine learning techniques to furnish the missing details. In particular, learning the behavioural specification of NSs is necessary.
5. **Dependability assurance:** In addition to the correctness properties expected from the synthesis under the assumption of a full NS description, it is also desirable to ensure that the CONNECTed system yields the appropriate non-functional properties such as dependability, security and trust.
6. **CONNECTing eternal systems:** Given the several enabling technologies involved in implementing the CONNECT approach, it is necessary to investigate the architecture of

the approach itself, with particular regard to the distribution and interaction between enablers in achieving the generation and maintenance of CONNECTors.

7. **Experiment:** The effectiveness of the overall approach will be shown through the integration of the various technologies and some assessment of the runtime efficiency.

The next section discusses the criteria that we consider for assessing the project results against the above objectives, again based on the Description of Work. Initial assessment is further provided, while it will be thoroughly elaborated in the next period and reported in D6.4.

5.2 Assessing CONNECT Results

5.2.1 Modelling and reasoning about peer system functionalities

A key aim of the CONNECT and CONNECTor architectures (WP1) is to both accurately model and reason about networked systems that wish to interoperate. In order to evaluate this overall aim, we will evaluate the effectiveness of the architecture against three underpinning objectives. Here we outline these objectives along with the evaluation methodology that will be applied to measure the success of the objective against the stated assessment criteria. For each, we also provide a preliminary assessment based upon the results achieved by the end of the third year of the project.

Objective 1: To accurately *model* the interfaces and functional behaviour provided and required by a given networked system.

Assessment criteria:

- The **accuracy** of the modelled affordances, interfaces and behaviour of a networked system in comparison to its actual interfaces and behaviour;
- The **flexibility** of CONNECT to model heterogeneous networked systems.

Methodology:

The CONNECT architecture will be applied to model the behaviour of the set of networked systems in the GMES scenarios. In certain cases the affordance is provided by the discovery protocol, in others the affordance must be learned. The produced Networked System model in each case will be compared to: the API published by the networked system, the networked system's behaviour, the middleware protocols employed by the networked system, and finally the non-functional requirements specified by the networked system. The results of this comparison will inform the accuracy assessment criteria.

As introduced in the previous chapters, the GMES scenario consists of a broad set of networked systems that exhibit heterogeneity in different dimensions, i.e., different application interfaces and behaviours, heterogeneous middleware interaction protocols, different discovery protocols and heterogeneous description languages. For example, the mobile weather station networked system uses the CDP discovery protocol with the Lime middleware protocol to discover and interact with other services, whereas the fixed camera networked systems uses WS-Discovery and the HTTP protocol. A qualitative analysis of the prior comparison method across all of the networked systems in the GMES scenario will be performed to assess the flexibility criteria.

Preliminary assessment:

The Discovery enabler and Learning enabler are the key elements in producing the networked system models. Their implementation has progressed sufficiently to test their operation on example networked systems. The Discovery enabler has been shown to discover networked systems advertised using different protocols (specifically CDP, WS-Discovery) employing different specification languages (WSDL, xDL); further

tests have shown that the Discovery enabler and Learning enabler can model the application behaviour of different networked systems employing different middleware protocols. Further, in the situations where the affordance is not provided, the initial results concerning affordance learning lean towards suitable accuracy being achieved. Hence, considering the assessment criteria, the initial results indicate promising progress towards achieving both the accuracy and flexibility required once applied to the full evaluation method.

Objective 2: To match two networked systems correctly.

Assessment criterion:

- The **accuracy of matches** that is the percentage of possible matches that are accurately determined, versus the number of matches missed and false positives.

Methodology:

The CONNECT architecture will be deployed and executed within the GMES scenario to discover and then match networked systems. The use cases within the scenario will include cases where goals are specified in the networked system model (and hence goal-based matching can take place), and cases where no goals are specified (and hence interface-based matching will be used). The successful matches will be identified, along with matches that were missed, and also any false positives (i.e., identified matches that were not appropriate for interoperation). These results will then be used to assess the accuracy of the matches, and evaluate the extent to which the CONNECT architecture can correctly match two networked systems.

Preliminary assessment:

A preliminary evaluation of both mapping-driven matching and goal-based matching has been performed for particular application domains (not the GMES scenario). Initial results indicate that, while matching is not always 100% accurate, the matching is reasonable and will add value to the execution of the GMES scenario.

Objective 3: To perform CONNECT networked system modelling and matching in a performant manner.

Assessment criterion:

- The **performance feasibility** that is the time taken to produce networked system models and match compatible networked systems in comparison to time required by legacy systems to have a match returned to them (e.g., the three second default in Service Location Protocol).

Methodology:

The CONNECT architecture will be deployed and executed to produce the networked system models networked systems in the GMES scenario. The time taken to perform each will be measured. The time taken for the CONNECT architecture to determine a match and then return this result to the requesting legacy networked system will also be measured. Finally, the required performance time of each legacy system will be analysed in terms of the wait time for a match response, and number of retries. Based upon these quantitative measures we will assess the performance feasibility of CONNECT according to networked systems in the GMES scenario.

Preliminary assessment:

Assessment of the performance of discovery, affordance learning and matching is in progress. Preliminary results show that affordance learning provides a substantial performance gain during the matchmaking process since fewer behavioural checks

need to be performed. The full experimental methodology will be performed in the final year of the project.

5.2.2 Modelling and reasoning about connector behaviours

The objective of WP2 is to provide a comprehensive theory to enable composition of connectors and automatically learn and reason about connector behaviours via a quantitative assume-guarantee reasoning paradigm. The expected outcomes are formalisms, methods and software tools that can be used for the specification, design and development of connectors, allowing for both functional and non-functional properties to be expressed and verified. We have developed and implemented quantitative assume-guarantee reasoning techniques in the last three years, and published the results in leading international conferences. Therefore, the success of this work package will be evaluated on the specification theory formalism, which we began to work on in Y2. There are two criteria for the evaluation.

Objective 1: Innovative formalism

Assessment criterion:

- Since there are many formalisms in the literature that address system modelling, we will assess our formalism on its **significance**, in the sense of advancing the state of the art, and the range of functionality of the operators that is supported. This can be judged by high quality publications, e.g., in leading international conferences and journals.

Methodology:

The evaluation will be based on assessment against state of the art and publications of the proposed formalisms in leading venues.

Preliminary assessment:

The specification theory that we introduce is influenced by interface automata, but relies on traces as opposed to the more complex alternating simulation. It supports composability of components at run-time and a refinement preorder that enables safe-substitutivity of components. The operators include parallel composition, conjunction and disjunction for independent development, and quotient for incremental development, as well as hiding.

The first paper about the specification theory has been accepted to ESOP 2012, a leading symposium on programming. Our specification theory addresses a number of shortcomings of interface automata and supports a broader range of operators than previously achieved. In particular, we propose the first quotient for non-deterministic interface automata.

Objective 2: Introducing the theoretical underpinning for emergent connectors

Assessment criterion:

- WP2 intends to provide the theoretical underpinning for the work carried out in the other work packages, in the sense that **connectors specified in WP2 can be instantiated** in WP3 (synthesis), WP4 (learning) and WP5 (dependability analysis). Thus a key challenge when developing the specification theory is to ensure that it is fully integrated and usable by WPs 3, 4 and 5.

Methodology:

We will assess the application of the formalism to the concrete approaches developed in WP3 to 5.

Preliminary assessment:

We have studied CONNECT-relevant examples where they have been available, for example, mediator synthesis from WP3. We have also developed an extension of the specification theory in D2.3 aimed at the register automata of WP4, but without imposing restrictions (determinism, canonicity) where this is not necessary. We have applied the quantitative assume-guarantee verification framework to the dependability analysis in WP5 (see D2.2 and D2.3).

We have demonstrated integration with WP3 in D2.3 by showing how quotient can be used for synthesis. In particular, we have shown that a mediator synthesised by WP3 is the most general mediator, and are able to check whether it is free from errors by using the theory.

5.2.3 Runtime synthesis of connectors

Regarding the work carried out within WP3 and concerning automated CONNECTor synthesis, the main challenges to be faced are: (i) to address the well-known protocol mismatches that are described in Deliverable D3.2; (ii) to perform automated CONNECTor synthesis on the fly and at runtime; (iii) to devise compositional approaches to the CONNECTor synthesis so as to support scalability and evolution of the synthesised CONNECTors; and (iv) to cope with QoS aspects of the considered NSs interaction behaviour, hence synthesising CONNECTors that satisfy specific non-functional requirements, i.e., that address CONNECTability dimensions as defined by WP5.

In order to evaluate how much the work done within WP3 addresses the above challenges, we will assess the developed CONNECTor synthesis methods and tools against the four objectives listed below. As done for the other sections of this chapter, we outline these objectives along with the applied evaluation methodology and related assessment criteria. For each objective, we also provide a preliminary assessment based upon the results achieved at the end of the third year of the project.

Objective 1: Overcoming a large range of protocol mismatches in an automated way.

Assessment criterion:

- The **common protocol mismatches** that may indeed be overcome by the automated synthesis of mediators, thereby enabling the connection of functionally matching systems.

Methodology:

The synthesis of mediators will be tested against the various types of NSs making part of the GMES scenario.

Synthesis may further be assessed theoretically according to base protocol mismatches, as discussed below.

Preliminary assessment:

As described in Deliverable D3.3, at present, the Synthesis enabler implements two CONNECTor synthesis methods, respectively referred to as the *goal-based* method and the *mapping-driven* method.

Given the large number of heterogeneous networked systems that can be encountered, we aim at having a toolbox within the Synthesis enabler for mediator synthesis featuring various approaches. Each approach has properties that make it better suited for specific cases. Table 4 evaluates the goal-based and mapping-driven synthesis methods against common mismatches:

- *Signature mismatch*: concerns actions with different naming, which is naturally treated in both approaches by considering the ontology-based semantics of actions.

- *Splitting of actions*: relates to having an input action of one system realised by a number of output actions of the other. Then, an input action may be split into a number of output actions of the matching networked system if such a relation holds from the domain-specific ontology. This is handled in both approaches.
- *Merging of actions*: defines an output action of one system that realises a number of input actions of the other. The goal-based synthesis solves it if the input actions do not require any return parameter, or use asynchronous invocations. This is necessary because, with a synchronous semantics, an action of the output sequence would be blocked waiting for its output, which in general will not come until the required action is performed. The mapping-driven synthesis handles it only in the case the input action does not require output data.
- *Extra send (or missing receive)*: During the synthesis of the mediator, both approaches ensure that any extra parameter, sent and not foreseen by the second party in communication, is consumed in order to avoid deadlock.
- *Extra receive (or missing send)*: a required action on one of the systems expects some input parameters that are not provided by the other party involved in the communication. Neither approach handles this mismatch as it violates the underlying notion of compatibility (see D3.3). In general, a solution to this kind of mismatches can be found if the missing parameters have default values that can be used to hide their absence. The current implementations proposed in D3.3 do not consider this case.
- *Ordering mismatch*: This concerns the re-ordering of actions so that networked systems may indeed co-ordinate. While the mapping-driven synthesis does not handle this case unless the networked systems models are made concurrent *a priori*, the goal-based one is able to manage this mismatch assuming asynchronous semantics.

	Goal-based Synthesis	Mapping-driven Synthesis
Signature mismatch	Yes	Yes
Splitting of actions	Yes	Yes
Merging of actions	Yes, asynchronous semantics needed	Only in the case input actions do not require output parameters
Extra output	Yes	Yes
Extra inputoutput actionsend	No	No
Ordering mismatch	Yes, if there is no extra output . Needs asynchronous semantics.	No
Goals	Yes	No
Complete CONNECTor produced	No, if eLTSs have loops	Yes

Table 4: Comparing goal-based and mapping-driven abstract CONNECTor synthesis

Furthermore, considering the goal used in the goal-based method, it allows the synthesis approach to support cases when there is only a partial match between the behaviour of the NSs. However, when such a goal is not provided, this approach needs to test several possible traces and collect them at the end. Therefore, the mapping-driven synthesis is well suited for the cases where the goal is not specified as it considers the whole behaviour of systems and checks that each possible execution of one system can possibly be mapped to an execution of the other system. Moreover, the mapping-driven synthesis can detect the impossibility of mediation while building the ontology-based action mapping thanks to the constraints that these mappings need to satisfy, and in this case avoid expensive behavioural checking. However, mapping-driven synthesis succeeds only if each input action has all its outputs available at time of occurrence, which is essential to prove the correctness of the mediator. Hence, many-to-many mismatches with asynchronous semantics and loops cannot be handled effectively. The goal-based approach, instead, bases its matching phase on a reachability problem on a counter transition system. This approach allows considering

both synchronous and asynchronous behaviours, loops and solves merging actions and reordering mismatches also in those cases disregarded by the mapping-driven approach. On the other hand, the goal-based approach is able to produce only subsets of the whole CONNECTor in those cases in which the interactions allowed by the networked system protocols are infinite.

Objective2: performing CONNECTor synthesis on the fly and at runtime.

Assessment criteria:

- The **efficiency of the CONNECTor synthesis** with respect to *time* and *resource consumption.*
- The ability to perform **CONNECTor synthesis on partial models/observations** of the interaction behaviour of the considered NSs.

Methodology:

The synthesis of mediators will be experimented against specific running instances of the GMES scenario.

Preliminary assessment:

By referring to the two aforementioned CONNECTor synthesis methods, the goal-based method, although still far from being applied on partial models, is more amenable than the mapping-driven one to be enhanced for the purposes of incomplete models. In fact, accounting for the goal specification allows the synthesis method to produce a goal-based opportunistic mediator that, among all the possible protocol matches that ensure NS interoperability, realizes only those that allow the CONNECTed system to exhibit at least one behaviour that fulfils the specified goal. Consequently, the computational complexity is generally lower than that of the mapping-driven method, which exhaustively tests all the execution traces. Nevertheless, extensive performance evaluation is necessary in order to better assess the applicability (and the efficiency) of the synthesis process at runtime. The goal is to clearly identify the time complexity and the resource usage of each step in the synthesis regarding the overall performance and considering both synthesis methods.

Objective 3: Devising compositional approaches to the synthesis of CONNECTors.

Assessment criterion:

- The ability to exploit as much as possible an already synthesized CONNECTor whenever **changes in the CONNECTed NSs** or **system goal** occur.

Methodology:

The synthesis of mediators will be experimented by changing the requirements, e.g., the goal to be achieved, of running instances of the GMES scenario that are already CONNECTed.

Preliminary assessment:

At present, the study of the *compositionality* dimension is still an ongoing work. We are working to enhance/adapt our developed methods and tools to cope with it. In particular, the CONNECTed system needs to evolve as new knowledge is being discovered or learned and to reflect changes in the operating environment. Therefore, the system is monitored continuously to identify executions that do not conform to the learned NS behaviour. This verification is carried out at runtime and the model of the mediator is updated accordingly so as to reflect the changes in the NS model. Ontologies may also evolve over time, although less frequently. In this case, the CONNECTed system is a closed-loop system to better deal with the partial knowledge it has about the environment. A remaining important challenge is to manage efficiently

the changes of the NSs models in order to re-synthesise the mediator in an incremental way.

Objective 4: Accounting for QoS aspects of the NSs interaction.

Assessment criterion:

- The **dependability, performance, and security metrics** devised by WP5.

Methodology:

The synthesis of mediators will be experimented against the CONNECTability attributes exhibited by the NSs participating to the GMES scenario. These attributes may correspond to a subset of the dependability, performance, and security metrics defined by WP5.

Preliminary assessment:

As described in Deliverable D3.2, so far, we have carried out a preliminary assessment of CONNECTor synthesis against security properties only. These properties essentially abstract sequences of allowed or legal (with respect to security) actions. They are expressed as logic formulae whose semantics can be suitably represented as automata or labelled transition systems, e.g., by using temporal logic notations. Note also that WP5 recently developed a property meta-model (see Deliverable D5.3) that can be exploited to represent these properties by using more practical notations while still keeping their mapping into automata-based notations. The preliminary evaluation that we did so far considers a theoretical method based on partial model-checking techniques as described in D3.2. Note that the goal-based synthesis method can be used as an alternative practical method whenever the goal specification represents the specification of a security property.

5.2.4 Learning connector behaviours

The overall goal of the Learning work is to support a bootstrapping mechanism which can start from minimal information about a networked system and generate useful behavioural models. The approach taken for learning behaviours will be assessed according to whether and how well it can take account of previous knowledge, and it will be assessed as well according to the quality of learned models and the range of aspects (data, non-functional properties) that can be covered. Assessment will also be according to whether and how well it can support evolution by means of techniques for incremental learning, involving detection of deviations from learned behaviour and triggering appropriate learning action. A non-negligible part of evaluation is done as part of evaluation of the overall CONNECT architecture (WP1), and observing how well the Learning Enabler can fulfil its role: see the evaluation plan for WP1.

Objective 1: Learning should be able exploit available information about a networked system, and should minimally need only a description of interfaces

Assessment Criteria:

- How much prior description is needed for learning.
- How well learning can exploit additional information.

Methodology:

Implemented Learning techniques will be applied to case studies provided throughout the period of CONNECT. These case studies provide the learning enabler with interface descriptions of different form and quality. It should be examined whether the learning enabler is able to successfully generate behaviour models with these descriptions as prior knowledge.

In some case studies, additional pre-processing may establish enhanced prior or derived knowledge. It should be assessed how such additional knowledge enhances the learning process and the generated models.

Preliminary Assessment:

Several case studies (see, e.g., Chapter 5 of D4.3) have demonstrated how the learning proceeds automatically from the availability of interface descriptions and information about where to direct invocations. Additionally, type information in interface descriptions can be used to enhance learning, and make learning more efficient. This is demonstrated in Chapter 4 of D4.3.

Objective 2: To generate accurate behaviour models of networked systems, covering a range of aspects

Assessment Criteria:

- How accurate are the learned models
- What range of aspects can be covered by learned models.

Methodology:

In case studies performed throughout CONNECT, the learned models will be assessed for accuracy. This will be done by several means, including conformance testing, post-learning monitoring, and also by assessing how well the models can be used in the CONNECT architecture.

In the case studies, it will also be assessed to what extent the learned models can capture data aspects of NS behaviour, as well whether bounds on timing properties are respected.

Preliminary Assessment:

A significant set of case studies have demonstrated that learning can produce sizeable and detailed models of systems. Data parameters can also be handled in learning; for simple examples this is demonstrated in Chapter 3 of D4.3.

Objective 3: To support evolution

Assessment Criterion:

- Whether and how well evolution can be detected, and how well learning updates models.

Methodology:

A case study will be devised in which a networked system is updated. It will be assessed how well this update is detected by, e.g., monitoring infrastructure, and how efficient the updating process can learn the evolved behaviour.

Preliminary Assessment:

The infrastructure to support evolution has been developed and experimented on simple examples, but not yet assessed on a full scenario.

5.2.5 Dependability assurance

The objectives for dependability assurance in CONNECT decompose into:

- A successful approach for dependability assurance in CONNECT will consist in having a set of clear qualitative and quantitative metrics to apply to CONNECTed systems such as security and privacy levels.
- The approach should automate the creation of validation suites for end-to-end monitoring of interactions, which can be measured by experimentation with the WP6 scenarios.

- The effectiveness of the approach will also depend on the robustness of the verification and validation techniques to such disruptions as system evolution, faults and malicious attacks.
- The approach should provide trust mechanisms that handle dynamic compositions and security policy languages with sufficient expressivity.

Objective 1: Qualitative and quantitative metrics

Assessment criterion:

- As stated in the DOW, the assessment will consider clearness, i.e. **lack of ambiguity and degree of formalisation**, of definition of CONNECT related non-functional properties. This is an important feature to support automated analysis (see Objective 2 below).

Methodology:

To evaluate the achievement of the above objective, we need to assess the capability and ease of use for prospective CONNECT users to refer to the provided formalism for specification of properties and metrics as an input to their analysis and assessment tools. The methodology for assessment will follow two directions: flexibility and breadth. In particular, the degree of success will be the higher the more flexible the specification is with respect to automated transformation into different formalisms, and the broader the range of properties that can be expressed.

Preliminary assessment:

At the end of Y3 we consider this objective is satisfied to a certain degree: to express the desired CONNECTability properties, both qualitative and quantitative ones (i.e. metrics), in a clear, formal way we have in fact developed the CONNECT Property Meta Model (CPMM), which has been introduced in D5.2 and has now been enriched in D5.3. The definition of CPMM has been carried out by taking into account related existing conceptual models and unifying, into one comprehensive framework, all aspects considered relevant for CONNECTed systems. Concerning flexibility, we can already use CPMM definitions as an input to the GLIMPSE monitor, by transforming CPMM into Drools Fusion. Other transformations will be considered in the fourth year. Concerning breadth, CPMM is now complete and expressive enough for defining dependability, performance and security properties, whereas we still need to complete and integrate it with concepts relating to trust properties.

Objective 2: Automation

Assessment criterion:

- **Degree of automation** in guiding and performing runtime validation of CONNECTed systems through monitoring

Methodology:

To evaluate the achievement of the above objective, we intend to provide experimental evidence through the CONNECT scenarios, showing demonstration as far as possible of a full story from user-desired non-functional properties down to runtime monitoring of those properties for a CONNECTed system, passing through all the steps of discovery, synthesis, analysis and deployment.

Preliminary assessment:

The implementation of a framework for allowing automated end-to-end monitoring is in good shape, as we have delivered in D5.3 a first version of a translator from CPMM to the monitor's native language, i.e. Drools Fusion: therefore once the desired non-functional properties are specified, the monitor engine can be automatically instructed about the events to be monitored via the probes inserted into the CONNECTor. It should

be noted however that properly speaking monitoring will not happen in general end-to-end: this would entail to be intrusive on the Networked Systems. This happens concerning specific security properties, but not for dependability and performance, in which case the monitoring involves the messages exchanged through the CONNECTor. The assessment through experimentation on WP6 scenarios has been provided so far only on some small examples both in D5.2 and D5.3, and we plan more extensive and integrated experimentation in the fourth year, once the set up of the GMES scenario is ready.

Objective 3: Robustness of V&V

Assessment criterion:

- **How robust are** proposed V&V frameworks to handle CONNECTion disruptions as system evolution, faults and malicious attacks.

Methodology:

To evaluate the achievement of the above objective, we intend to provide experimental evidence demonstrating within the CONNECT scenarios how DePer can adapt the analysis in presence of system evolution and accidental faults, and how the Security Enabler can resist malicious attacks, by enforcing the defined security policies.

Preliminary assessment:

Concerning the assessment of the robustness of V&V techniques against disruptions, we have secured the integration between the DePer Enabler and the GLIMPSE monitor, as described in D5.2, so that the latter can, in a timely manner, detect possible events that are a signal of such issues and inform DePer to revise V&V analyses. During the fourth year, integration with the Synthesis Enabler will be also finalised, concerning dependability measures to be accommodated in the CONNECTor synthesis to cope with disruptions. We have also interfaced the security enabler with the monitor. We have not yet shown a real integrated case study; in Y4, based on the achieved implementation of the GMES scenario, we will show a more comprehensive case study.

Objective 4: Expressing trust and security

Assessment criteria:

- Capability for trust management to **support dynamic composition**
- Capability for security enabler to **handle relevant security** policies.

Methodology:

We intend to assess the successful satisfaction of both above requirements by experimental evaluation of both the trust management system and the security enabler on the GMES scenario.

Preliminary assessment:

Trust mechanisms to handle dynamic composition between heterogeneous Networked Systems have been formally defined in D5.3 and preliminarily assessed in the current year on an example taken from the GMES scenario. A simple XML-based security policy language for CONNECT has been introduced in D5.3, following a standard approach for expressing security and trust policies. It has been demonstrated so far on a small example, taken again from the GMES scenario. We intend to conduct further assessment in Y4 regarding the adequacy in expressiveness for trust and security frameworks.

5.2.6 Connecting eternal systems

The principle output of the CONNECT project is the CONNECT architecture; the aims of the overall architecture are to i) achieve the principle project goal, namely universal, long-lived, and future-proof interoperability, and ii) to provide breakthrough solutions that impact upon the field of distributed systems. In order to assess the architecture against these overall goals we will evaluate the effectiveness of the architecture against three underpinning objectives.

Objective 1: To utilise the CONNECT architecture to automatically generate and deploy concrete CONNECTors that successfully ensure correct interoperation between two matched networked systems.

Assessment criteria:

- The **correctness of** a **CONNECTor;** the functionality of the CONNECTor ensures each networked system operates according to its original requirements;
- The **performant behaviour** of the generated CONNECTor.

Methodology:

The CONNECT architecture will be applied to the GMES scenario, i.e., the architecture will be deployed into the scenario in order to automatically generate a CONNECTOR in the required cases. A qualitative evaluation will then be performed to analyse each of the produced CONNECTors to determine the degree to which they are correct according to the functional and non-functional requirements of the deployed networked systems. Further, a quantitative evaluation of the behaviour of each CONNECTor will measure the time taken when the CONNECTor is used to perform interoperation; this measure will be compared to the expected performance requirements of the networked system.

Preliminary assessment:

The concrete architecture is incomplete and hence no preliminary assessment can be provided concerning the overall CONNECT behaviour. However, numerous experiments to produce CONNECTors to resolve heterogeneity indicate that there has been progress towards achieving this objective and the results indicate that this can be achieved and demonstrated in the GMES scenario. Protocol bridges have been generated to CONNECT heterogeneous middleware protocols (SOAP to CORBA to XML-RPC). Application mediators have been automatically produced for different application domains exhibiting application heterogeneity. Further, in one case (the photo-sharing example described in D1.3) a CONNECTor has been created to resolve combined application and middleware heterogeneity.

Objective 2: To correctly adapt a deployed CONNECTor when the non-functional requirements are validated or the initial CONNECTor is incorrect with respect to the functional requirements.

Assessment criteria:

- The **correctness of adaptations**; that is, adaptations are performed at the correct time and improve upon the prior state;
- The **degree of disruption**, i.e., the extent to which the overall operation of the networked systems is disrupted by the CONNECT architecture performing adaptations.

Methodology:

Within the GMES scenario, a set of non-functional use cases concern situations where the CONNECTor must be adapted due to the analysis of the CONNECTability enablers (i.e. DePer, Security, and Trust) indicating that the current CONNECTor behaviour violates a particular non-functional requirement. The CONNECT architecture will be executed for each of the use-cases, and a qualitative evaluation will assess the

correctness of this adaptation with respect to the behaviour provided before and after by the mediator (against the required non-functional properties). Quantitative measures of the execution of each use-case will be performed to measure any failures in the networked systems and the downtime of the CONNECTor during adaptation. The quantitative measure will be used to assess the degree of disruption introduced by the CONNECT architecture and the positive or negative effect this has on the overall system behaviour.

Preliminary assessment:

The integration of adaptation into the CONNECT architecture is in progress; the implementation will be completed in the fourth year and the evaluation methodology previously described will be utilised to assess the success of achieving correct adaptations within the whole architecture.

Objective 3: For the technologies and solutions created within the CONNECT to have a broader impact in the field of distributed systems; and for the wider use of CONNECT software to resolve interoperability problems.

Assessment criterion:

- **Uptake of principles, solutions and technologies** proposed and developed by CONNECT.

Methodology:

The evaluation will employ two methods to analyse the effective uptake of the project output. The first will utilise statistical measures of software usage (downloads), and project references (e.g. citations to CONNECT papers). The second will analyse anecdotal evidence of the usage of CONNECT, e.g. the specific usage of CONNECT software in other software projects, or the usage of CONNECT deployments to build interoperable solutions.

Preliminary assessment:

No measures have so far been produced about the uptake of CONNECT output and hence no preliminary assessment can be made.

5.2.7 Experiment

WP6 will provide scenarios consisting of a wide range of heterogeneous systems that challenge the various features of the CONNECT architecture. In particular, the GMES scenario from Thales will provide various networked systems with differing affordances, interaction protocols, and dependability requirements, e.g., with regard to security and trust properties. The case studies provided by Ambientic presents a different set of challenges for the CONNECT architecture, which relates to its exploitation for mobile, collaborative applications.

While the previous sections have focused on assessing CONNECT solutions from a generic perspective, this section concentrates on domain-specific assessment of CONNECT solutions, considering more specifically the domain of Systems of Systems.

As defined in Wikipedia's page on Systems of Systems[1], several traits are inherent to System of Systems:

- Operational Independence of Elements;
- Managerial Independence of Elements;
- Evolutionary Development;
- Emergent Behaviour;

[1] http://en.wikipedia.org/wiki/System_of_systems

- Geographical Distribution of Elements;
- Inter-disciplinary Study;
- Heterogeneity of Systems; and
- Networks of Systems

While some of them are not relevant to CONNECT (such as Inter-disciplinary Study), the others have been gathered in four main objectives, which are to be assessed, given the commonalities of their assessment methodology:

- Operational and Managerial Independence of Systems;
- Evolutionary Development and Emergent Behaviour;
- Heterogeneity of Systems; and
- Networks of Systems and Geographical Distribution of Elements.

Orthogonally to these objectives, the overall performance, especially with respect to the overhead of the CONNECT platform (globally and per enabler) will be studied.

Objective 1: To respect the Operational and Managerial Independence of Systems.

Assessment criterion:

- The non-intrusiveness of the CONNECT platform on the networked systems.

Methodology:

Within the GMES scenario, the evaluation will consist of checking the capability to modify certain networked systems (code, configuration files…) and the capability of the CONNECT platform to accommodate these modifications, without interfering with the rest of a CONNECTed System of Systems.

Preliminary assessment:

No measure has been produced so far.

Objective 2: To allow the Evolutionary Development and Emergent Behaviour.

Assessment criteria:

- Support of Evolutionary development;
- Detection of Emergent Behaviour.

Methodology:

Within the GMES scenario, the evaluation will consist of:

- Adding and removing Networked Systems to and from the System of Systems, taking specific care of adding and removing Systems that involve different CONNECTors;
- Evolving the code of the C2 system in order to command in a different way the Networked Systems, hence establishing a new, emergent, behaviour.

Preliminary assessment:

At this time, the only issue raised as of today is that we cannot express compositions of affordances to represent a composite service. For example a drone with an embedded fixed camera will be seen as a drone plus a fixed camera (two affordances) and it is impossible to expose a moving and rotating camera resulting of the merge/ composition of the camera on the mobile platform. A future support of this composition of affordances might be a hint for Emergent Behaviour detection.

Objective 3: To support the Heterogeneity of Systems.

Assessment criteria:

- The number of supported exchange patterns;

- The number of transport protocols supported;
- The ease for these exchange patterns and transport protocols to evolve.

Methodology:

Within the GMES scenario a set of exchange patterns and associated transport protocols have already been selected. The evaluation will so consist of checking their effective realisation and integration when executing the scenario.

Ambientic's video stream adaptation will also be a key contributor to this objective.

Preliminary assessment:

No measure has been produced so far.

Objective 4: Networks of Systems and Geographical Distribution of Elements.

Assessment criteria:

- The density of the networks of systems supported;
- The effective distributiveness of the networked systems in a CONNECT architecture.

Methodology:

The evaluation of the first criterion will consist of running various alternative ways to start the networked systems (gradually, by set, all at the same time) with a large number of CONNECTed mock-ups of networked systems and see how the System of Systems react.

Regarding the second criterion, the weather service of the demonstration is to be available on the Internet and we will also attempt to integrate cameras deployed at different locations.

Preliminary assessment:

No measure has been produced so far.

5.3 Summary

Assessment of the objectives for individual enablers has in most cases been performed with respect to various different case studies. This has shown that the objectives have indeed been achieved to a certain degree, and what remains for the fourth year is to gauge how effective each enabler is in tackling the GMES scenario, and how successfully each enabler operates in the overall architecture.

One particular overall objective is for reasonable runtime performance in connecting two networked systems, which has an impact upon each enabler in the synthesis pipeline. A further area of focus for Y4 will be runtime feedback and consequent adaptation, which also concerns every enabler and the dependencies between them.

6 CONCLUSION

In Year 3 of CONNECT, the partners in WP6 have continued their work toward providing a suitable platform for assessing the research performed in WP1 – WP5 in a realistic setting. Specifically, based on the feedback received in the previous review, we now have the definition of a comprehensive GMES system, which now includes many different devices and allows us to experiment with different forms of interoperability as we mix various interaction paradigms, transport protocols and data formats. This also paves the way for illustrating the underlying architecture and the CONNECT enablers in action, thus demonstrating the overall CONNECT approach.

WP6 has collaborated closely with WP1 and all other WPs to produce an implemented CONNECT System; as stated in Chapter 4, the initial versions of some of the NSs are already available, and the CONNECT enabler implementations are rapidly progressing towards a level of maturity. We expect to be able to produce and assess a first version of the demonstrator in the coming months.

Indeed thanks to the prior works done on PTZ cameras during last year we are able to provide rapidly the realization of the three kinds of camera (fixed, main and patrol), and also to have rapid progress on the specifications of other involved networked systems, both in terms of interfaces and affordances. The fire-fighter videoconference NS is of special interest, since it addresses streaming of both audio and video on the network; its specifications will be provided in D6.4.

Finally, we have made progress towards the use of semantic technologies in CONNECT, resulting in the draft version of the domain-specific ontology dedicated to the demonstration. This showcases the research conducted in WP1 on discovery and affordance matching in WP3, and will further evolve as we perform assessment on the implemented NSs of the CONNECT enablers.

7 APPENDIX

7.1 NS 1.1 - UAV

7.1.1 XML Schema

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="http://drone.connect.arles.inria.fr/"
xmlns:tns="http://drone.connect.arles.inria.fr/" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="authenticate" type="tns:authenticate"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVAuthargs"/>

  <xs:element name="authenticateResponse" type="tns:authenticateResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVAccessTokenString"/>

  <xs:element name="getCoordinates" type="tns:getCoordinates"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVAccessToken"/>

  <xs:element name="getCoordinatesResponse" type="tns:getCoordinatesResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVCoordinates"/>

  <xs:element name="land" type="tns:land"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVAccessToken"/>

  <xs:element name="landResponse" type="tns:landResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVResponse"/>

  <xs:element name="logout" type="tns:logout"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVAccessToken"/>

  <xs:element name="logoutResponse" type="tns:logoutResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVResponse"/>

  <xs:element name="moveback" type="tns:moveback"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVAccessTokenWithTime"/>

  <xs:element name="movebackResponse" type="tns:movebackResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVResponse"/>

  <xs:element name="movedown" type="tns:movedown"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVAccessTokenWithTime"/>

  <xs:element name="movedownResponse" type="tns:movedownResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVResponse"/>

  <xs:element name="moveforward" type="tns:moveforward"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVAccessTokenWithTime "/>

  <xs:element name="moveforwardResponse" type="tns:moveforwardResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVResponse"/>

  <xs:element name="moveleft" type="tns:moveleft"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVAccessTokenWithTime"/>

  <xs:element name="moveleftResponse" type="tns:moveleftResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVResponse"/>

  <xs:element name="moveright" type="tns:moveright"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVAccessTokenWithTime"/>

  <xs:element name="moverightResponse" type="tns:moverightResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVResponse"/>

  <xs:element name="moveup" type="tns:moveup"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVAccessTokenWithTime"/>

  <xs:element name="moveupResponse" type="tns:moveupResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVResponse"/>

  <xs:element name="takeoff" type="tns:takeoff"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVAccessTokenWithTime"/>
```

```
  <xs:element name="takeoffResponse" type="tns:takeoffResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVResponse"/>

  <xs:complexType name="logout">
    <xs:sequence>
      <xs:element name="accesstoken" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="logoutResponse">
    <xs:sequence>
      <xs:element name="return" type="tns:droneResponse" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="droneResponse">
    <xs:sequence>
      <xs:element name="status" type="tns:status" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="takeoff">
    <xs:sequence>
      <xs:element name="accesstoken" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="takeoffResponse">
    <xs:sequence>
      <xs:element name="return" type="tns:droneResponse" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="getCoordinates">
    <xs:sequence>
      <xs:element name="accesstoken" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="getCoordinatesResponse">
    <xs:sequence>
      <xs:element name="return" type="tns:coordinatesResponse" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="coordinatesResponse">
    <xs:sequence>
      <xs:element name="payload" type="tns:coordinates" minOccurs="0"/>
      <xs:element name="status" type="tns:status" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="coordinates">
    <xs:sequence>
      <xs:element name="pitch" type="xs:double"/>
      <xs:element name="roll" type="xs:double"/>
      <xs:element name="x" type="xs:double"/>
      <xs:element name="y" type="xs:double"/>
      <xs:element name="yaw" type="xs:double"/>
      <xs:element name="z" type="xs:double"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="authenticate">
    <xs:sequence>
      <xs:element name="login" type="xs:string" minOccurs="0"/>
      <xs:element name="password" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="authenticateResponse">
    <xs:sequence>
      <xs:element name="return" type="tns:stringResponse" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
```

```
<xs:complexType name="stringResponse">
  <xs:sequence>
    <xs:element name="payload" type="xs:string" minOccurs="0"/>
    <xs:element name="status" type="tns:status" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="moveback">
  <xs:sequence>
    <xs:element name="accesstoken" type="xs:string" minOccurs="0"/>
    <xs:element name="duration" type="xs:long"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="movebackResponse">
  <xs:sequence>
    <xs:element name="return" type="tns:droneResponse" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="movedown">
  <xs:sequence>
    <xs:element name="accesstoken" type="xs:string" minOccurs="0"/>
    <xs:element name="duration" type="xs:long"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="movedownResponse">
  <xs:sequence>
    <xs:element name="return" type="tns:droneResponse" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="moveforward">
  <xs:sequence>
    <xs:element name="accesstoken" type="xs:string" minOccurs="0"/>
    <xs:element name="duration" type="xs:long"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="moveforwardResponse">
  <xs:sequence>
    <xs:element name="return" type="tns:droneResponse" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="moveup">
  <xs:sequence>
    <xs:element name="accesstoken" type="xs:string" minOccurs="0"/>
    <xs:element name="duration" type="xs:long"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="moveupResponse">
  <xs:sequence>
    <xs:element name="return" type="tns:droneResponse" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="moveright">
  <xs:sequence>
    <xs:element name="accesstoken" type="xs:string" minOccurs="0"/>
    <xs:element name="duration" type="xs:long"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="moverightResponse">
  <xs:sequence>
    <xs:element name="return" type="tns:droneResponse" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="land">
  <xs:sequence>
    <xs:element name="accesstoken" type="xs:string" minOccurs="0"/>
```

```
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="landResponse">
    <xs:sequence>
      <xs:element name="return" type="tns:droneResponse" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="moveleft">
    <xs:sequence>
      <xs:element name="accesstoken" type="xs:string" minOccurs="0"/>
      <xs:element name="duration" type="xs:long"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="moveleftResponse">
    <xs:sequence>
      <xs:element name="return" type="tns:droneResponse" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:simpleType name="status">
    <xs:restriction base="xs:string">
      <xs:enumeration value="error"/>
      <xs:enumeration value="success"/>
      <xs:enumeration value="droneError"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

7.1.2 SAWSDL

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is
        JAX-WS RI 2.2.1-hudson-28-. -->
<definitions targetNamespace="http://drone.connect.arles.inria.fr/"
        name="DroneService" xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsp="http://www.w3.org/ns/ws-policy"
        xmlns:tns="http://drone.connect.arles.inria.fr/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
        xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
        xmlns:sawsdl="http://www.w3.org/2007/01/sawsdl"
        xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
        <types>
                <xsd:schema>
                        <xsd:import namespace="http://drone.connect.arles.inria.fr/"
                                schemaLocation="DroneService_schema1.xsd" />
                </xsd:schema>
        </types>
        <message name="authenticate">
                <part name="parameters" element="tns:authenticate" />
        </message>
        <message name="authenticateResponse">
                <part name="parameters" element="tns:authenticateResponse" />
        </message>
        <message name="movedown">
                <part name="parameters" element="tns:movedown" />
        </message>
        <message name="movedownResponse">
                <part name="parameters" element="tns:movedownResponse" />
        </message>
        <message name="moveup">
                <part name="parameters" element="tns:moveup" />
        </message>
        <message name="moveupResponse">
                <part name="parameters" element="tns:moveupResponse" />
        </message>
        <message name="land">
```

```
                <part name="parameters" element="tns:land" />
        </message>
        <message name="landResponse">
                <part name="parameters" element="tns:landResponse" />
        </message>
        <message name="logout">
                <part name="parameters" element="tns:logout" />
        </message>
        <message name="logoutResponse">
                <part name="parameters" element="tns:logoutResponse" />
        </message>
        <message name="getCoordinates">
                <part name="parameters" element="tns:getCoordinates" />
        </message>
        <message name="getCoordinatesResponse">
                <part name="parameters" element="tns:getCoordinatesResponse" />
        </message>
        <message name="takeoff">
                <part name="parameters" element="tns:takeoff" />
        </message>
        <message name="takeoffResponse">
                <part name="parameters" element="tns:takeoffResponse" />
        </message>
        <message name="moveleft">
                <part name="parameters" element="tns:moveleft" />
        </message>
        <message name="moveleftResponse">
                <part name="parameters" element="tns:moveleftResponse" />
        </message>
        <message name="moveright">
                <part name="parameters" element="tns:moveright" />
        </message>
        <message name="moverightResponse">
                <part name="parameters" element="tns:moverightResponse" />
        </message>
        <message name="moveforward">
                <part name="parameters" element="tns:moveforward" />
        </message>
        <message name="moveforwardResponse">
                <part name="parameters" element="tns:moveforwardResponse" />
        </message>
        <message name="moveback">
                <part name="parameters" element="tns:moveback" />
        </message>
        <message name="movebackResponse">
                <part name="parameters" element="tns:movebackResponse" />
        </message>
        <portType name="DroneService">
                <operation name="authenticate"
sawsdl:modelReference="http://www.connect.com/ontology/media##UAVLogin">
                        <input

        wsam:Action="http://drone.connect.arles.inria.fr/DroneService/authenticateRequest"
                                message="tns:authenticate" />
                        <output

        wsam:Action="http://drone.connect.arles.inria.fr/DroneService/authenticateResponse"
                                message="tns:authenticateResponse" />
                </operation>
                <operation name="movedown"
sawsdl:modelReference="http://www.connect.com/ontology/media##UAVMoveDown">
                        <input

        wsam:Action="http://drone.connect.arles.inria.fr/DroneService/movedownRequest"
                                message="tns:movedown" />
                        <output

        wsam:Action="http://drone.connect.arles.inria.fr/DroneService/movedownResponse"
                                message="tns:movedownResponse" />
                </operation>
                <operation name="moveup"
sawsdl:modelReference="http://www.connect.com/ontology/media##UAVMoveUp">
                        <input

        wsam:Action="http://drone.connect.arles.inria.fr/DroneService/moveupRequest"
                                message="tns:moveup" />
```

```
                        <output

        wsam:Action="http://drone.connect.arles.inria.fr/DroneService/moveupResponse"
                                message="tns:moveupResponse" />
                </operation>
                <operation name="land"
sawsdl:modelReference="http://www.connect.com/ontology/media##UAVLand">
                        <input
wsam:Action="http://drone.connect.arles.inria.fr/DroneService/landRequest"
                                message="tns:land" />
                        <output

        wsam:Action="http://drone.connect.arles.inria.fr/DroneService/landResponse"
                                message="tns:landResponse" />
                </operation>
                <operation name="logout"
sawsdl:modelReference="http://www.connect.com/ontology/media##UAVLogout">
                        <input

        wsam:Action="http://drone.connect.arles.inria.fr/DroneService/logoutRequest"
                                message="tns:logout" />
                        <output

        wsam:Action="http://drone.connect.arles.inria.fr/DroneService/logoutResponse"
                                message="tns:logoutResponse" />
                </operation>
                <operation name="getCoordinates"
sawsdl:modelReference="http://www.connect.com/ontology/media##UAVGetCoordinates">
                        <input

        wsam:Action="http://drone.connect.arles.inria.fr/DroneService/getCoordinatesRequest"
                                message="tns:getCoordinates" />
                        <output

        wsam:Action="http://drone.connect.arles.inria.fr/DroneService/getCoordinatesResponse"
                                message="tns:getCoordinatesResponse" />
                </operation>
                <operation name="takeoff"
sawsdl:modelReference="http://www.connect.com/ontology/media##UAVTakeoff">
                        <input

        wsam:Action="http://drone.connect.arles.inria.fr/DroneService/takeoffRequest"
                                message="tns:takeoff" />
                        <output

        wsam:Action="http://drone.connect.arles.inria.fr/DroneService/takeoffResponse"
                                message="tns:takeoffResponse" />
                </operation>
                <operation name="moveleft"
sawsdl:modelReference="http://www.connect.com/ontology/media##UAVMoveLeft">
                        <input

        wsam:Action="http://drone.connect.arles.inria.fr/DroneService/moveleftRequest"
                                message="tns:moveleft" />
                        <output

        wsam:Action="http://drone.connect.arles.inria.fr/DroneService/moveleftResponse"
                                message="tns:moveleftResponse" />
                </operation>
                <operation name="moveright"
sawsdl:modelReference="http://www.connect.com/ontology/media##UAVMoveRight">
                        <input

        wsam:Action="http://drone.connect.arles.inria.fr/DroneService/moverightRequest"
                                message="tns:moveright" />
                        <output

        wsam:Action="http://drone.connect.arles.inria.fr/DroneService/moverightResponse"
                                message="tns:moverightResponse" />
                </operation>
                <operation name="moveforward"
sawsdl:modelReference="http://www.connect.com/ontology/media##UAVMoveForward">
                        <input

        wsam:Action="http://drone.connect.arles.inria.fr/DroneService/moveforwardRequest"
                                message="tns:moveforward" />
```

```
                        <output

        wsam:Action="http://drone.connect.arles.inria.fr/DroneService/moveforwardResponse"
                                message="tns:moveforwardResponse" />
                </operation>
                <operation name="moveback"
sawsdl:modelReference="http://www.connect.com/ontology/media##UAVMoveBack">
                        <input

        wsam:Action="http://drone.connect.arles.inria.fr/DroneService/movebackRequest"
                                message="tns:moveback" />
                        <output

        wsam:Action="http://drone.connect.arles.inria.fr/DroneService/movebackResponse"
                                message="tns:movebackResponse" />
                </operation>
        </portType>
        <binding name="DroneServicePortBinding" type="tns:DroneService">
                <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
                        style="document" />
                <operation name="authenticate">
                        <soap:operation soapAction="" />
                        <input>
                                <soap:body use="literal" />
                        </input>
                        <output>
                                <soap:body use="literal" />
                        </output>
                </operation>
                <operation name="movedown">
                        <soap:operation soapAction="" />
                        <input>
                                <soap:body use="literal" />
                        </input>
                        <output>
                                <soap:body use="literal" />
                        </output>
                </operation>
                <operation name="moveup">
                        <soap:operation soapAction="" />
                        <input>
                                <soap:body use="literal" />
                        </input>
                        <output>
                                <soap:body use="literal" />
                        </output>
                </operation>
                <operation name="land">
                        <soap:operation soapAction="" />
                        <input>
                                <soap:body use="literal" />
                        </input>
                        <output>
                                <soap:body use="literal" />
                        </output>
                </operation>
                <operation name="logout">
                        <soap:operation soapAction="" />
                        <input>
                                <soap:body use="literal" />
                        </input>
                        <output>
                                <soap:body use="literal" />
                        </output>
                </operation>
                <operation name="getCoordinates">
                        <soap:operation soapAction="" />
                        <input>
                                <soap:body use="literal" />
                        </input>
                        <output>
                                <soap:body use="literal" />
                        </output>
                </operation>
                <operation name="takeoff">
                        <soap:operation soapAction="" />
```

```
                        <input>
                                <soap:body use="literal" />
                        </input>
                        <output>
                                <soap:body use="literal" />
                        </output>
                </operation>
                <operation name="moveleft">
                        <soap:operation soapAction="" />
                        <input>
                                <soap:body use="literal" />
                        </input>
                        <output>
                                <soap:body use="literal" />
                        </output>
                </operation>
                <operation name="moveright">
                        <soap:operation soapAction="" />
                        <input>
                                <soap:body use="literal" />
                        </input>
                        <output>
                                <soap:body use="literal" />
                        </output>
                </operation>
                <operation name="moveforward">
                        <soap:operation soapAction="" />
                        <input>
                                <soap:body use="literal" />
                        </input>
                        <output>
                                <soap:body use="literal" />
                        </output>
                </operation>
                <operation name="moveback">
                        <soap:operation soapAction="" />
                        <input>
                                <soap:body use="literal" />
                        </input>
                        <output>
                                <soap:body use="literal" />
                        </output>
                </operation>
        </binding>
        <service name="DroneService">
                <port name="DroneServicePort" binding="tns:DroneServicePortBinding">
                        <soap:address location="REPLACE_WITH_ACTUAL_URL" />
                </port>
        </service>
</definitions>
```

7.2 NS 1.2 - UAV Camera

7.2.1 XML Schema

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="http://drone.connect.arles.inria.fr/"
xmlns:tns="http://drone.connect.arles.inria.fr/" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="authenticate" type="tns:authenticate"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVAuthargs"/>

  <xs:element name="authenticateResponse" type="tns:authenticateResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVAccessTokenString"/>

  <xs:element name="authorizeStream" nillable="true" type="tns:authorizeStream"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVAccessTokenWithStreamId"/>

  <xs:element name="authorizeStreamResponse" nillable="true" type="tns:authorizeStreamResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVStreamUrl"/>

  <xs:element name="getStreamClients" nillable="true" type="tns:getStreamClients"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVAccessTokenWithStreamId"/>
```

```
  <xs:element name="getStreamClientsResponse" nillable="true"
type="tns:getStreamClientsResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVClientsList"/>

  <xs:element name="getStreamDescriptions" nillable="true" type="tns:getStreamDescriptions"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVAccessToken"/>

  <xs:element name="getStreamDescriptionsResponse" nillable="true"
type="tns:getStreamDescriptionsResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVStreamDescriptionList"/>

  <xs:element name="logout" nillable="true" type="tns:logout"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVAccessToken"/>

  <xs:element name="logoutResponse" nillable="true" type="tns:logoutResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#UAVResponse"/>

  <xs:complexType name="getStreamDescriptions">
    <xs:sequence>
      <xs:element name="accesstoken" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="getStreamDescriptionsResponse">
    <xs:sequence>
      <xs:element name="return" type="tns:streamDescriptionListResponse" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="streamDescriptionListResponse">
    <xs:sequence>
      <xs:element name="payload" type="tns:streamDescription" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="status" type="tns:status" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="streamDescription">
    <xs:sequence>
      <xs:element name="framerate" type="xs:int" minOccurs="0"/>
      <xs:element name="name" type="xs:string" minOccurs="0"/>
      <xs:element name="protocol" type="xs:string" minOccurs="0"/>
      <xs:element name="resolution" type="xs:string" minOccurs="0"/>
      <xs:element name="type" type="xs:string" minOccurs="0"/>
      <xs:element name="uri" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="authenticate">
    <xs:sequence>
      <xs:element name="login" type="xs:string" minOccurs="0"/>
      <xs:element name="password" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="authenticateResponse">
    <xs:sequence>
      <xs:element name="return" type="tns:stringResponse" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="stringResponse">
    <xs:sequence>
      <xs:element name="payload" type="xs:string" minOccurs="0"/>
      <xs:element name="status" type="tns:status" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="logout">
    <xs:sequence>
      <xs:element name="accesstoken" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="logoutResponse">
```

```
    <xs:sequence>
      <xs:element name="return" type="tns:droneResponse" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="droneResponse">
    <xs:sequence>
      <xs:element name="status" type="tns:status" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="authorizeStream">
    <xs:sequence>
      <xs:element name="accesstoken" type="xs:string" minOccurs="0"/>
      <xs:element name="sourcename" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="authorizeStreamResponse">
    <xs:sequence>
      <xs:element name="return" type="tns:stringResponse" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="getStreamClients">
    <xs:sequence>
      <xs:element name="accesstoken" type="xs:string" minOccurs="0"/>
      <xs:element name="sourcename" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="getStreamClientsResponse">
    <xs:sequence>
      <xs:element name="return" type="tns:stringListResponse" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="stringListResponse">
    <xs:sequence>
      <xs:element name="payload" type="xs:string" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="status" type="tns:status" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:simpleType name="status">
    <xs:restriction base="xs:string">
      <xs:enumeration value="error"/>
      <xs:enumeration value="success"/>
      <xs:enumeration value="droneError"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

7.2.2 SAWSDL

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2.1-
hudson-28-. -->
<definitions targetNamespace="http://drone.connect.arles.inria.fr/" name="DroneStreamService"
xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:tns="http://drone.connect.arles.inria.fr/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <types>
    <xsd:schema>
      <xsd:import namespace="http://drone.connect.arles.inria.fr/"
schemaLocation="DroneStreamService_schema1.xsd"/>
    </xsd:schema>
  </types>
```

```
  <message name="authenticate">
    <part name="parameters" element="tns:authenticate"/>
  </message>
  <message name="authenticateResponse">
    <part name="parameters" element="tns:authenticateResponse"/>
  </message>
  <message name="getStreamDescriptions">
    <part name="parameters" element="tns:getStreamDescriptions"/>
  </message>
  <message name="getStreamDescriptionsResponse">
    <part name="parameters" element="tns:getStreamDescriptionsResponse"/>
  </message>
  <message name="getStreamClients">
    <part name="parameters" element="tns:getStreamClients"/>
  </message>
  <message name="getStreamClientsResponse">
    <part name="parameters" element="tns:getStreamClientsResponse"/>
  </message>
  <message name="authorizeStream">
    <part name="parameters" element="tns:authorizeStream"/>
  </message>
  <message name="authorizeStreamResponse">
    <part name="parameters" element="tns:authorizeStreamResponse"/>
  </message>
  <message name="logout">
    <part name="parameters" element="tns:logout"/>
  </message>
  <message name="logoutResponse">
    <part name="parameters" element="tns:logoutResponse"/>
  </message>
  <portType name="DroneStreamService">
    <operation name="authenticate"
sawsdl:modelReference="http://www.connect.com/ontology/media##UAVLogin">
      <input
wsam:Action="http://drone.connect.arles.inria.fr/DroneStreamService/authenticateRequest"
message="tns:authenticate"/>
      <output
wsam:Action="http://drone.connect.arles.inria.fr/DroneStreamService/authenticateResponse"
message="tns:authenticateResponse"/>
    </operation>
    <operation name="getStreamDescriptions"
sawsdl:modelReference="http://www.connect.com/ontology/media##UAVGetStreamDescriptions">
      <input
wsam:Action="http://drone.connect.arles.inria.fr/DroneStreamService/getStreamDescriptionsReques
t" message="tns:getStreamDescriptions"/>
      <output
wsam:Action="http://drone.connect.arles.inria.fr/DroneStreamService/getStreamDescriptionsRespon
se" message="tns:getStreamDescriptionsResponse"/>
    </operation>
    <operation name="getStreamClients"
sawsdl:modelReference="http://www.connect.com/ontology/media##UAVGetStreamClients">
      <input
wsam:Action="http://drone.connect.arles.inria.fr/DroneStreamService/getStreamClientsRequest"
message="tns:getStreamClients"/>
      <output
wsam:Action="http://drone.connect.arles.inria.fr/DroneStreamService/getStreamClientsResponse"
message="tns:getStreamClientsResponse"/>
    </operation>
    <operation name="authorizeStream"
sawsdl:modelReference="http://www.connect.com/ontology/media##UAVAuthorizeStream">
      <input
wsam:Action="http://drone.connect.arles.inria.fr/DroneStreamService/authorizeStreamRequest"
message="tns:authorizeStream"/>
      <output
wsam:Action="http://drone.connect.arles.inria.fr/DroneStreamService/authorizeStreamResponse"
message="tns:authorizeStreamResponse"/>
    </operation>
    <operation name="logout"
sawsdl:modelReference="http://www.connect.com/ontology/media##UAVLogout">
      <input wsam:Action="http://drone.connect.arles.inria.fr/DroneStreamService/logoutRequest"
message="tns:logout"/>
      <output
wsam:Action="http://drone.connect.arles.inria.fr/DroneStreamService/logoutResponse"
message="tns:logoutResponse"/>
    </operation>
  </portType>
```

```
  <binding name="DroneStreamServicePortBinding" type="tns:DroneStreamService">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <operation name="authenticate">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
    <operation name="getStreamDescriptions">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
    <operation name="getStreamClients">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
    <operation name="authorizeStream">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
    <operation name="logout">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="DroneStreamService">
    <port name="DroneStreamServicePort" binding="tns:DroneStreamServicePortBinding">
      <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
    </port>
  </service>
</definitions>
```

7.3 NS 2 - UGV

7.3.1 SAWSDL

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
        xmlns:tns="http://services.connect.thalesgroup.com/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="UGVehicle"
        targetNamespace="http://services.connect.thalesgroup.com/"
xmlns:sawsdl="http://www.w3.org/2007/01/sawsdl#">
        <wsdl:types>
                <xsd:schema targetNamespace="http://services.connect.thalesgroup.com/">
                        <xsd:element name="setMovingPattern"
sawsdl:modelReference="http://www.connect.com/ontology/media#MovePattern">
                                <xsd:complexType>
                                        <xsd:sequence>
```

```
                                        <xsd:element name="pattern" type="tns:patternType"
/>
                                </xsd:sequence>
                        </xsd:complexType>
                </xsd:element>
                <xsd:element name="setMovingPatternResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#Void">
                        <xsd:complexType>
                                <xsd:sequence>
                                </xsd:sequence>
                        </xsd:complexType>
                </xsd:element>
                <xsd:element name="move"
sawsdl:modelReference="http://www.connect.com/ontology/media#MoveState">
                        <xsd:complexType>
                                <xsd:sequence>
                                        <xsd:element name="state"
type="xsd:boolean"></xsd:element>
                                </xsd:sequence>
                        </xsd:complexType>
                </xsd:element>
                <xsd:element name="moveResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#Void">
                        <xsd:complexType>
                                <xsd:sequence>
                                </xsd:sequence>
                        </xsd:complexType>
                </xsd:element>
                <xsd:element name="zoom"
sawsdl:modelReference="http://www.connect.com/ontology/media#ZoomFactor">
                        <xsd:complexType>
                                <xsd:sequence>
                                        <xsd:element name="ratio"
type="xsd:int"></xsd:element>
                                </xsd:sequence>
                        </xsd:complexType>
                </xsd:element>
                <xsd:element name="zoomResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#Void">
                        <xsd:complexType>
                                <xsd:sequence>
                                </xsd:sequence>
                        </xsd:complexType>
                </xsd:element>
                <xsd:element name="panAndTilt"
sawsdl:modelReference="http://www.connect.com/ontology/media#PanTiltParameters">
                        <xsd:complexType>
                                <xsd:sequence>
                                        <xsd:element name="pan"
type="xsd:double"></xsd:element>
                                        <xsd:element name="tilt"
type="xsd:double"></xsd:element>
                                </xsd:sequence>
                        </xsd:complexType>
                </xsd:element>
                <xsd:element name="panAndTiltResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#Void">
                        <xsd:complexType>
                                <xsd:sequence>
                                </xsd:sequence>
                        </xsd:complexType>
                </xsd:element>
                <xsd:element name="getVideoMJPEGAddress"
sawsdl:modelReference="http://www.connect.com/ontology/media#Void">
                        <xsd:complexType>
                                <xsd:sequence>
                                </xsd:sequence>
                        </xsd:complexType>
                </xsd:element>
                <xsd:element name="getVideoMJPEGAddressResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#URL">
                        <xsd:complexType>
                                <xsd:sequence>
                                        <xsd:element name="out"
type="xsd:string"></xsd:element>
                                </xsd:sequence>
```

```
                        </xsd:complexType>
                    </xsd:element>
                    <xsd:simpleType name="patternType">
                        <xsd:restriction base="xsd:string">
                            <xsd:enumeration value="CIRCLE" />
                            <xsd:enumeration value="SQUARE" />
                            <xsd:enumeration value="DIAMOND" />
                        </xsd:restriction>
                    </xsd:simpleType>
                    <xsd:element name="connectPrim">
                        <xsd:complexType>
                            <xsd:sequence>
                                <xsd:element name="in"
type="xsd:string"></xsd:element>
                            </xsd:sequence>
                        </xsd:complexType>
                    </xsd:element>
                    <xsd:element name="NewOperationResponse">
                        <xsd:complexType>
                            <xsd:sequence>

                                <xsd:element name="out"
type="xsd:string"></xsd:element>
                            </xsd:sequence>
                        </xsd:complexType>
                    </xsd:element>
                    <xsd:element name="connectPrimResponse">
                        <xsd:complexType>
                            <xsd:sequence>

                                <xsd:element name="out"
type="xsd:string"></xsd:element>
                            </xsd:sequence>
                        </xsd:complexType>
                    </xsd:element>
                </xsd:schema>
        </wsdl:types>
        <wsdl:message name="setMovingPatternRequest">
                <wsdl:part element="tns:setMovingPattern" name="parameters" />
        </wsdl:message>
        <wsdl:message name="setMovingPatternResponse">
                <wsdl:part element="tns:setMovingPatternResponse" name="parameters" />
        </wsdl:message>
        <wsdl:message name="moveRequest">
                <wsdl:part name="parameters" element="tns:move"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="moveResponse">
                <wsdl:part name="parameters" element="tns:moveResponse"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="zoomRequest">
                <wsdl:part name="parameters" element="tns:zoom"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="zoomResponse">
                <wsdl:part name="parameters" element="tns:zoomResponse"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="panAndTiltRequest">
                <wsdl:part name="parameters" element="tns:panAndTilt"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="panAndTiltResponse">
                <wsdl:part name="parameters" element="tns:panAndTiltResponse"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="getVideoMJPEGAddressRequest">
                <wsdl:part name="parameters" element="tns:getVideoMJPEGAddress"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="getVideoMJPEGAddressResponse">
                <wsdl:part name="parameters"
element="tns:getVideoMJPEGAddressResponse"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="connectPrimRequest">
                <wsdl:part name="parameters" element="tns:connectPrim"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="NewOperationResponse">
                <wsdl:part name="parameters" element="tns:NewOperationResponse"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="connectPrimResponse1">
                <wsdl:part name="parameters" element="tns:connectPrimResponse"></wsdl:part>
```

```
        </wsdl:message>
        <wsdl:portType name="UGVehicle">
                <wsdl:operation name="setMovingPattern"
sawsdl:modelReference="http://www.connect.com/ontology/media#SetMovePattern">
                        <wsdl:input message="tns:setMovingPatternRequest" />
                        <wsdl:output message="tns:setMovingPatternResponse" />
                </wsdl:operation>
                <wsdl:operation name="move"
sawsdl:modelReference="http://www.connect.com/ontology/media#Move">
                        <wsdl:input message="tns:moveRequest"></wsdl:input>
                        <wsdl:output message="tns:moveResponse"></wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="zoom"
sawsdl:modelReference="http://www.connect.com/ontology/media#Zoom">
                        <wsdl:input message="tns:zoomRequest"></wsdl:input>
                        <wsdl:output message="tns:zoomResponse"></wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="panAndTilt"
                        sawsdl:modelReference="http://www.connect.com/ontology/media#PanTilt">
                        <wsdl:input message="tns:panAndTiltRequest"></wsdl:input>
                        <wsdl:output message="tns:panAndTiltResponse"></wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="getVideoMJPEGAddress"
sawsdl:modelReference="http://www.connect.com/ontology/media#ShowVideo">
                        <wsdl:input message="tns:getVideoMJPEGAddressRequest"></wsdl:input>
                        <wsdl:output message="tns:getVideoMJPEGAddressResponse"></wsdl:output>
                </wsdl:operation>

                <wsdl:operation name="connectPrim">
                        <wsdl:input message="tns:connectPrimRequest"></wsdl:input>
                        <wsdl:output message="tns:connectPrimResponse1"></wsdl:output>
                </wsdl:operation>
        </wsdl:portType>
        <wsdl:binding name="UGVehicleSOAP" type="tns:UGVehicle">
                <soap:binding style="document"
                        transport="http://schemas.xmlsoap.org/soap/http" />
                <wsdl:operation name="setMovingPattern">
                        <soap:operation

        soapAction="http://services.connect.thalesgroup.com/setMovingPattern" />
                        <wsdl:input>
                                <soap:body use="literal" />
                        </wsdl:input>
                        <wsdl:output>
                                <soap:body use="literal" />
                        </wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="move">
                        <soap:operation
                                soapAction="http://services.connect.thalesgroup.com/move" />
                        <wsdl:input>
                                <soap:body use="literal" />
                        </wsdl:input>
                        <wsdl:output>
                                <soap:body use="literal" />
                        </wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="zoom">
                        <soap:operation
                                soapAction="http://services.connect.thalesgroup.com/zoom" />
                        <wsdl:input>
                                <soap:body use="literal" />
                        </wsdl:input>
                        <wsdl:output>
                                <soap:body use="literal" />
                        </wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="panAndTilt">
                        <soap:operation
                                soapAction="http://services.connect.thalesgroup.com/panAndTilt"
/>
                        <wsdl:input>
                                <soap:body use="literal" />
                        </wsdl:input>
                        <wsdl:output>
                                <soap:body use="literal" />
```

```
                        </wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="getVideoMJPEGAddress">
                        <soap:operation

        soapAction="http://services.connect.thalesgroup.com/getVideoMJPEGAddress" />
                        <wsdl:input>
                                <soap:body use="literal" />
                        </wsdl:input>
                        <wsdl:output>
                                <soap:body use="literal" />
                        </wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="connectPrim">
                        <soap:operation
                                soapAction="http://services.connect.thalesgroup.com/connectPrim"
/>
                        <wsdl:input>
                                <soap:body use="literal" />
                        </wsdl:input>
                        <wsdl:output>
                                <soap:body use="literal" />
                        </wsdl:output>
                </wsdl:operation>
        </wsdl:binding>
        <wsdl:service name="UGVehicle">
                <wsdl:port binding="tns:UGVehicleSOAP" name="UGVehicleSOAP">
                        <soap:address location="http://www.example.org/" />
                </wsdl:port>
        </wsdl:service>
</wsdl:definitions>
```

7.4 NS 3.1 - Camera Fixed

7.4.1 SAWSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
        xmlns:tns="http://services.connect.thalesgroup.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://services.connect.thalesgroup.com/"
        name="CameraFixedService" xmlns:sawsdl="http://www.w3.org/2007/01/sawsdl#">
        <types>
                <xs:schema version="1.0"
                        targetNamespace="http://services.connect.thalesgroup.com/"
xmlns:tns="http://services.connect.thalesgroup.com/"
                        xmlns:xs="http://www.w3.org/2001/XMLSchema">
                        <xs:element name="connectPrim" type="tns:connectPrim"></xs:element>
                        <xs:element name="connectPrimResponse"
type="tns:connectPrimResponse"></xs:element>
                        <xs:element name="getVideoRTSPAddress" type="tns:getVideoRTSPAddress"

        sawsdl:modelReference="http://www.connect.com/ontology/media#Void"></xs:element>
                        <xs:element name="getVideoRTSPAddressResponse"
type="tns:getVideoRTSPAddressResponse"

        sawsdl:modelReference="http://www.connect.com/ontology/media#RTSPAddress"></xs:element>
                        <xs:element name="zoom" type="tns:zoom"

        sawsdl:modelReference="http://www.connect.com/ontology/media#ZoomFactor"></xs:element>
                        <xs:element name="zoomResponse" type="tns:zoomResponse"

        sawsdl:modelReference="http://www.connect.com/ontology/media#Void"></xs:element>
                        <xs:complexType name="getVideoRTSPAddress">
                                <xs:sequence></xs:sequence>
                        </xs:complexType>
                        <xs:complexType name="getVideoRTSPAddressResponse">
                                <xs:sequence>
                                        <xs:element name="return" type="xs:string"
minOccurs="0"></xs:element>
                                </xs:sequence>
```

```
                        </xs:complexType>
                        <xs:complexType name="zoom">
                                <xs:sequence>
                                        <xs:element name="ratio" type="xs:int"></xs:element>
                                </xs:sequence>
                        </xs:complexType>
                        <xs:complexType name="zoomResponse">
                                <xs:sequence></xs:sequence>
                        </xs:complexType>
                        <xs:complexType name="connectPrim">
                                <xs:sequence>
                                        <xs:element name="connectInput" type="xs:string"
                                                minOccurs="0"></xs:element>
                                </xs:sequence>
                        </xs:complexType>
                        <xs:complexType name="connectPrimResponse">
                                <xs:sequence>
                                        <xs:element name="return" type="xs:string"
minOccurs="0"></xs:element>
                                </xs:sequence>
                        </xs:complexType>
                </xs:schema>
        </types>
        <message name="zoom">
                <part name="parameters" element="tns:zoom"></part>
        </message>
        <message name="zoomResponse">
                <part name="parameters" element="tns:zoomResponse"></part>
        </message>
        <message name="getVideoRTSPAddress">
                <part name="parameters" element="tns:getVideoRTSPAddress"></part>
        </message>
        <message name="getVideoRTSPAddressResponse">
                <part name="parameters" element="tns:getVideoRTSPAddressResponse"></part>
        </message>
        <message name="connectPrim">
                <part name="parameters" element="tns:connectPrim"></part>
        </message>
        <message name="connectPrimResponse">
                <part name="parameters" element="tns:connectPrimResponse"></part>
        </message>
        <portType name="CameraFixed">
                <operation name="zoom"
sawsdl:modelReference="http://www.connect.com/ontology/media#Zoom">
                        <input message="tns:zoom"></input>
                        <output message="tns:zoomResponse"></output>
                </operation>
                <operation name="getVideoRTSPAddress"
                        sawsdl:modelReference="http://www.connect.com/ontology/media#ShowVideo">
                        <input message="tns:getVideoRTSPAddress"></input>
                        <output message="tns:getVideoRTSPAddressResponse"></output>
                </operation>
                <operation name="connectPrim">
                        <input message="tns:connectPrim"></input>
                        <output message="tns:connectPrimResponse"></output>
                </operation>
        </portType>
        <binding name="CameraFixedPortBinding" type="tns:CameraFixed">
                <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
                        style="document"></soap:binding>
                <operation name="zoom">
                        <soap:operation soapAction=""></soap:operation>
                        <input>
                                <soap:body use="literal"></soap:body>
                        </input>
                        <output>
                                <soap:body use="literal"></soap:body>
                        </output>
                </operation>
                <operation name="getVideoRTSPAddress">
                        <soap:operation soapAction=""></soap:operation>
                        <input>
                                <soap:body use="literal"></soap:body>
                        </input>
                        <output>
                                <soap:body use="literal"></soap:body>
```

```
                        </output>
                </operation>
                <operation name="connectPrim">
                        <soap:operation soapAction=""></soap:operation>
                        <input>
                                <soap:body use="literal"></soap:body>
                        </input>
                        <output>
                                <soap:body use="literal"></soap:body>
                        </output>
                </operation>
        </binding>
        <service name="CameraFixedService">
                <port name="CameraFixedPort" binding="tns:CameraFixedPortBinding">
                        <soap:address location="http://localhost/cameraFixed"></soap:address>
                </port>
        </service>
</definitions>
```

7.5 NS 3.2 - Camera Mobile Main

7.5.1 SAWSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
        xmlns:tns="http://services.connect.thalesgroup.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://services.connect.thalesgroup.com/"
        name="CameraMobileMainService" xmlns:sawsdl="http://www.w3.org/2007/01/sawsdl#">
        <types>
                <xs:schema version="1.0"
                        targetNamespace="http://services.connect.thalesgroup.com/"
xmlns:tns="http://services.connect.thalesgroup.com/"
                        xmlns:xs="http://www.w3.org/2001/XMLSchema">
                        <xs:element name="connectPrim" type="tns:connectPrim"></xs:element>
                        <xs:element name="connectPrimResponse"
type="tns:connectPrimResponse"></xs:element>
                        <xs:element name="getVideoMJPEGAddress" type="tns:getVideoMJPEGAddress"

        sawsdl:modelReference="http://www.connect.com/ontology/media#Void"></xs:element>
                        <xs:element name="getVideoMJPEGAddressResponse"
type="tns:getVideoMJPEGAddressResponse"

        sawsdl:modelReference="http://www.connect.com/ontology/media#MJPEGAddress"></xs:element>
                        <xs:element name="panAndTilt" type="tns:panAndTilt"

        sawsdl:modelReference="http://www.connect.com/ontology/media#PanTiltParameters"></xs:ele
ment>
                        <xs:element name="panAndTiltResponse" type="tns:panAndTiltResponse"

        sawsdl:modelReference="http://www.connect.com/ontology/media#Void"></xs:element>
                        <xs:element name="zoom" type="tns:zoom"

        sawsdl:modelReference="http://www.connect.com/ontology/media#ZoomFactor"></xs:element>
                        <xs:element name="zoomResponse" type="tns:zoomResponse"

        sawsdl:modelReference="http://www.connect.com/ontology/media#Void"></xs:element>
                        <xs:complexType name="panAndTilt">
                                <xs:sequence>
                                        <xs:element name="pan" type="xs:float"></xs:element>
                                        <xs:element name="tilt" type="xs:float"></xs:element>
                                </xs:sequence>
                        </xs:complexType>
                        <xs:complexType name="panAndTiltResponse">
                                <xs:sequence></xs:sequence>
                        </xs:complexType>
                        <xs:complexType name="zoom">
                                <xs:sequence>
                                        <xs:element name="ratio" type="xs:int"></xs:element>
                                </xs:sequence>
                        </xs:complexType>
```

```
                        <xs:complexType name="zoomResponse">
                                <xs:sequence></xs:sequence>
                        </xs:complexType>
                        <xs:complexType name="connectPrim">
                                <xs:sequence>
                                        <xs:element name="connectInput" type="xs:string"
                                                minOccurs="0"></xs:element>
                                </xs:sequence>
                        </xs:complexType>
                        <xs:complexType name="connectPrimResponse">
                                <xs:sequence>
                                        <xs:element name="return" type="xs:string"
minOccurs="0"></xs:element>
                                </xs:sequence>
                        </xs:complexType>
                        <xs:complexType name="getVideoMJPEGAddress">
                                <xs:sequence></xs:sequence>
                        </xs:complexType>
                        <xs:complexType name="getVideoMJPEGAddressResponse">
                                <xs:sequence>
                                        <xs:element name="return" type="xs:string"
minOccurs="0"></xs:element>
                                </xs:sequence>
                        </xs:complexType>
                </xs:schema>
        </types>
        <message name="zoom">
                <part name="parameters" element="tns:zoom"></part>
        </message>
        <message name="zoomResponse">
                <part name="parameters" element="tns:zoomResponse"></part>
        </message>
        <message name="connectPrim">
                <part name="parameters" element="tns:connectPrim"></part>
        </message>
        <message name="connectPrimResponse">
                <part name="parameters" element="tns:connectPrimResponse"></part>
        </message>
        <message name="getVideoMJPEGAddress">
                <part name="parameters" element="tns:getVideoMJPEGAddress"></part>
        </message>
        <message name="getVideoMJPEGAddressResponse">
                <part name="parameters" element="tns:getVideoMJPEGAddressResponse"></part>
        </message>
        <message name="panAndTilt">
                <part name="parameters" element="tns:panAndTilt"></part>
        </message>
        <message name="panAndTiltResponse">
                <part name="parameters" element="tns:panAndTiltResponse"></part>
        </message>
        <portType name="CameraMobileMain">
                <operation name="zoom"
sawsdl:modelReference="http://www.connect.com/ontology/media#Zoom">
                        <input message="tns:zoom"></input>
                        <output message="tns:zoomResponse"></output>
                </operation>
                <operation name="connectPrim">
                        <input message="tns:connectPrim"></input>
                        <output message="tns:connectPrimResponse"></output>
                </operation>
                <operation name="getVideoMJPEGAddress"
                        sawsdl:modelReference="http://www.connect.com/ontology/media#ShowVideo">
                        <input message="tns:getVideoMJPEGAddress"></input>
                        <output message="tns:getVideoMJPEGAddressResponse"></output>
                </operation>
                <operation name="panAndTilt"
                        sawsdl:modelReference="http://www.connect.com/ontology/media#PanTilt">
                        <input message="tns:panAndTilt"></input>
                        <output message="tns:panAndTiltResponse"></output>
                </operation>
        </portType>
        <binding name="CameraMobileMainPortBinding" type="tns:CameraMobileMain">
                <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
                        style="document"></soap:binding>
                <operation name="zoom">
                        <soap:operation soapAction=""></soap:operation>
```

```
                        <input>
                                <soap:body use="literal"></soap:body>
                        </input>
                        <output>
                                <soap:body use="literal"></soap:body>
                        </output>
                </operation>
                <operation name="connectPrim">
                        <soap:operation soapAction=""></soap:operation>
                        <input>
                                <soap:body use="literal"></soap:body>
                        </input>
                        <output>
                                <soap:body use="literal"></soap:body>
                        </output>
                </operation>
                <operation name="getVideoMJPEGAddress">
                        <soap:operation soapAction=""></soap:operation>
                        <input>
                                <soap:body use="literal"></soap:body>
                        </input>
                        <output>
                                <soap:body use="literal"></soap:body>
                        </output>
                </operation>
                <operation name="panAndTilt">
                        <soap:operation soapAction=""></soap:operation>
                        <input>
                                <soap:body use="literal"></soap:body>
                        </input>
                        <output>
                                <soap:body use="literal"></soap:body>
                        </output>
                </operation>
        </binding>
        <service name="CameraMobileMainService">
                <port name="CameraMobileMainPort" binding="tns:CameraMobileMainPortBinding">
                        <soap:address location="http://localhost/cameraMain"></soap:address>
                </port>
        </service>
</definitions>
```

7.6 NS 3.3 - Camera Mobile Patrol

7.6.1 SAWSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
        xmlns:tns="http://services.connect.thalesgroup.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://services.connect.thalesgroup.com/"
        name="CameraMobilePatrolService" xmlns:sawsdl="http://www.w3.org/2007/01/sawsdl#">
        <types>
                <xs:schema version="1.0"
                        targetNamespace="http://services.connect.thalesgroup.com/"
xmlns:tns="http://services.connect.thalesgroup.com/"
                        xmlns:xs="http://www.w3.org/2001/XMLSchema">
                        <xs:element name="connectPrim" type="tns:connectPrim"></xs:element>
                        <xs:element name="connectPrimResponse"
type="tns:connectPrimResponse"></xs:element>
                        <xs:element name="getTimedRecordedVideoLocalization"
type="tns:getTimedRecordedVideoLocalization"

        sawsdl:modelReference="http://www.connect.com/ontology/media#VideoParameters"></xs:eleme
nt>
                        <xs:element name="getTimedRecordedVideoLocalizationResponse"
                                type="tns:getTimedRecordedVideoLocalizationResponse"

        sawsdl:modelReference="http://www.connect.com/ontology/media#URL"></xs:element>
                        <xs:element name="startPatrol" type="tns:startPatrol"
```

```
        sawsdl:modelReference="http://www.connect.com/ontology/media#Void"></xs:element>
                        <xs:element name="startPatrolResponse" type="tns:startPatrolResponse"

        sawsdl:modelReference="http://www.connect.com/ontology/media#Void"></xs:element>
                        <xs:element name="stopPatrol" type="tns:stopPatrol"

        sawsdl:modelReference="http://www.connect.com/ontology/media#Void"></xs:element>
                        <xs:element name="stopPatrolResponse" type="tns:stopPatrolResponse"

        sawsdl:modelReference="http://www.connect.com/ontology/media#Void"></xs:element>
                        <xs:element name="zoom" type="tns:zoom"

        sawsdl:modelReference="http://www.connect.com/ontology/media#ZoomFactor"></xs:element>
                        <xs:element name="zoomResponse" type="tns:zoomResponse"

        sawsdl:modelReference="http://www.connect.com/ontology/media#Void"></xs:element>
                        <xs:complexType name="startPatrol">
                                <xs:sequence></xs:sequence>
                        </xs:complexType>
                        <xs:complexType name="startPatrolResponse">
                                <xs:sequence></xs:sequence>
                        </xs:complexType>
                        <xs:complexType name="stopPatrol">
                                <xs:sequence></xs:sequence>
                        </xs:complexType>
                        <xs:complexType name="stopPatrolResponse">
                                <xs:sequence></xs:sequence>
                        </xs:complexType>
                        <xs:complexType name="getTimedRecordedVideoLocalization">
                                <xs:sequence>
                                        <xs:element name="seconds" type="xs:int"></xs:element>
                                </xs:sequence>
                        </xs:complexType>
                        <xs:complexType name="getTimedRecordedVideoLocalizationResponse">
                                <xs:sequence>
                                        <xs:element name="return" type="xs:string"
minOccurs="0"></xs:element>
                                </xs:sequence>
                        </xs:complexType>
                        <xs:complexType name="zoom">
                                <xs:sequence>
                                        <xs:element name="ratio" type="xs:int"></xs:element>
                                </xs:sequence>
                        </xs:complexType>
                        <xs:complexType name="zoomResponse">
                                <xs:sequence></xs:sequence>
                        </xs:complexType>
                        <xs:complexType name="connectPrim">
                                <xs:sequence>
                                        <xs:element name="connectInput" type="xs:string"
                                                minOccurs="0"></xs:element>
                                </xs:sequence>
                        </xs:complexType>
                        <xs:complexType name="connectPrimResponse">
                                <xs:sequence>
                                        <xs:element name="return" type="xs:string"
minOccurs="0"></xs:element>
                                </xs:sequence>
                        </xs:complexType>
                </xs:schema>
        </types>
        <message name="zoom">
                <part name="parameters" element="tns:zoom"></part>
        </message>
        <message name="zoomResponse">
                <part name="parameters" element="tns:zoomResponse"></part>
        </message>
        <message name="connectPrim">
                <part name="parameters" element="tns:connectPrim"></part>
        </message>
        <message name="connectPrimResponse">
                <part name="parameters" element="tns:connectPrimResponse"></part>
        </message>
        <message name="getTimedRecordedVideoLocalization">
                <part name="parameters" element="tns:getTimedRecordedVideoLocalization"></part>
```

```
        </message>
        <message name="getTimedRecordedVideoLocalizationResponse">
                <part name="parameters"
element="tns:getTimedRecordedVideoLocalizationResponse"></part>
        </message>
        <message name="startPatrol">
                <part name="parameters" element="tns:startPatrol"></part>
        </message>
        <message name="startPatrolResponse">
                <part name="parameters" element="tns:startPatrolResponse"></part>
        </message>
        <message name="stopPatrol">
                <part name="parameters" element="tns:stopPatrol"></part>
        </message>
        <message name="stopPatrolResponse">
                <part name="parameters" element="tns:stopPatrolResponse"></part>
        </message>
        <portType name="CameraMobilePatrol">
                <operation name="zoom"
sawsdl:modelReference="http://www.connect.com/ontology/media#Zoom">
                        <input message="tns:zoom"></input>
                        <output message="tns:zoomResponse"></output>
                </operation>
                <operation name="connectPrim">
                        <input message="tns:connectPrim"></input>
                        <output message="tns:connectPrimResponse"></output>
                </operation>
                <operation name="getTimedRecordedVideoLocalization"

        sawsdl:modelReference="http://www.connect.com/ontology/media#RecordVideo">
                        <input message="tns:getTimedRecordedVideoLocalization"></input>
                        <output message="tns:getTimedRecordedVideoLocalizationResponse"></output>
                </operation>
                <operation name="startPatrol"

        sawsdl:modelReference="http://www.connect.com/ontology/media#StartPatrol">
                        <input message="tns:startPatrol"></input>
                        <output message="tns:startPatrolResponse"></output>
                </operation>
                <operation name="stopPatrol">
                        <input message="tns:stopPatrol"

        sawsdl:modelReference="http://www.connect.com/ontology/media#StopPatrol"></input>
                        <output message="tns:stopPatrolResponse"></output>
                </operation>
        </portType>
        <binding name="CameraMobilePatrolPortBinding" type="tns:CameraMobilePatrol">
                <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
                        style="document"></soap:binding>
                <operation name="zoom">
                        <soap:operation soapAction=""></soap:operation>
                        <input>
                                <soap:body use="literal"></soap:body>
                        </input>
                        <output>
                                <soap:body use="literal"></soap:body>
                        </output>
                </operation>
                <operation name="connectPrim">
                        <soap:operation soapAction=""></soap:operation>
                        <input>
                                <soap:body use="literal"></soap:body>
                        </input>
                        <output>
                                <soap:body use="literal"></soap:body>
                        </output>
                </operation>
                <operation name="getTimedRecordedVideoLocalization">
                        <soap:operation soapAction=""></soap:operation>
                        <input>
                                <soap:body use="literal"></soap:body>
                        </input>
                        <output>
                                <soap:body use="literal"></soap:body>
                        </output>
                </operation>
```

```
		<operation name="startPatrol">
			<soap:operation soapAction=""></soap:operation>
			<input>
				<soap:body use="literal"></soap:body>
			</input>
			<output>
				<soap:body use="literal"></soap:body>
			</output>
		</operation>
		<operation name="stopPatrol">
			<soap:operation soapAction=""></soap:operation>
			<input>
				<soap:body use="literal"></soap:body>
			</input>
			<output>
				<soap:body use="literal"></soap:body>
			</output>
		</operation>
	</binding>
	<service name="CameraMobilePatrolService">
		<port name="CameraMobilePatrolPort" binding="tns:CameraMobilePatrolPortBinding">
			<soap:address location="http://localhost/cameraPatrol"></soap:address>
		</port>
	</service>
</definitions>
```

7.7 NS 4 - System C2 GIS (client)

7.7.1 SAWSDL

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
	xmlns:tns="http://services.connect.thalesgroup.com/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
	xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="C2Gis"
	targetNamespace="http://services.connect.thalesgroup.com/"
	xmlns:sawsdl="http://www.w3.org/2007/01/sawsdl#">
	<wsdl:types>
		<xsd:schema targetNamespace="http://services.connect.thalesgroup.com/">
			<xsd:element name="getVideo"

	sawsdl:modelReference="http://www.connect.com/ontology/media#Void">
				<xsd:complexType>
					<xsd:sequence>
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
			<xsd:element name="getVideoResponse"

	sawsdl:modelReference="http://www.connect.com/ontology/media#URL">
				<xsd:complexType>
					<xsd:sequence>
						<xsd:element name="out" type="xsd:string" />
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
			<xsd:element name="panAndTilt"

	sawsdl:modelReference="http://www.connect.com/ontology/media#PanTiltParameters">
				<xsd:complexType>
					<xsd:sequence>
						<xsd:element name="pan"
type="xsd:double"></xsd:element>
						<xsd:element name="tilt"
type="xsd:double"></xsd:element>
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
			<xsd:element name="panAndTiltResponse"

	sawsdl:modelReference="http://www.connect.com/ontology/media#Void">
				<xsd:complexType>
					<xsd:sequence>
```

```
                                        </xsd:sequence>
                                </xsd:complexType>
                        </xsd:element>
                        <xsd:element name="zoom"

        sawsdl:modelReference="http://www.connect.com/ontology/media#ZoomFactor">
                                <xsd:complexType>
                                        <xsd:sequence>
                                                <xsd:element name="in"
type="xsd:string"></xsd:element>
                                        </xsd:sequence>
                                </xsd:complexType>
                        </xsd:element>
                        <xsd:element name="zoomResponse"

        sawsdl:modelReference="http://www.connect.com/ontology/media#Void">
                                <xsd:complexType>
                                        <xsd:sequence>
                                        </xsd:sequence>
                                </xsd:complexType>
                        </xsd:element>
                        <xsd:element name="getWeather"

        sawsdl:modelReference="http://www.connect.com/ontology/media#ZipCode">
                                <xsd:complexType>
                                        <xsd:sequence></xsd:sequence>
                                </xsd:complexType>
                        </xsd:element>
                        <xsd:element name="getWeatherResponse"

        sawsdl:modelReference="http://www.connect.com/ontology/media#WeatherData">
                                <xsd:complexType>
                                        <xsd:sequence>
                                                <xsd:element name="out"
type="xsd:string"></xsd:element>
                                        </xsd:sequence>
                                </xsd:complexType>
                        </xsd:element>
                        <xsd:element name="subscribePositioningSystem"

        sawsdl:modelReference="http://www.connect.com/ontology/media#Void">
                                <xsd:complexType>
                                </xsd:complexType>
                        </xsd:element>
                        <xsd:element name="subscribePositioningSystemResponse"

        sawsdl:modelReference="http://www.connect.com/ontology/media#Void">
                                <xsd:complexType></xsd:complexType>
                        </xsd:element>
                        <xsd:element name="moveVehicle"
sawsdl:modelReference="http://www.connect.com/ontology/media#MovePattern">
                                <xsd:complexType>
                                        <xsd:sequence>
                                                <xsd:element name="pattern"
type="xsd:string"></xsd:element>
                                        </xsd:sequence>
                                </xsd:complexType>
                        </xsd:element>
                        <xsd:element name="moveVehicleResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#Void">
                                <xsd:complexType>
                                        <xsd:sequence></xsd:sequence>
                                </xsd:complexType>
                        </xsd:element>
                </xsd:schema>
        </wsdl:types>
        <wsdl:message name="getVideoRequest">
                <wsdl:part element="tns:getVideo" name="parameters" />
        </wsdl:message>
        <wsdl:message name="getVideoResponse">
                <wsdl:part element="tns:getVideoResponse" name="parameters" />
        </wsdl:message>
        <wsdl:message name="panAndTiltRequest">
                <wsdl:part name="parameters" element="tns:panAndTilt"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="panAndTiltResponse">
```

```
                <wsdl:part name="parameters" element="tns:panAndTiltResponse"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="zoomRequest">
                <wsdl:part name="parameters" element="tns:zoom"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="zoomResponse">
                <wsdl:part name="parameters" element="tns:zoomResponse"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="getWeatherRequest">
                <wsdl:part name="parameters" element="tns:getWeather"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="getWeatherResponse">
                <wsdl:part name="parameters" element="tns:getWeatherResponse"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="subscribePositioningSystemRequest">
                <wsdl:part name="parameters"
element="tns:subscribePositioningSystem"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="subscribePositioningSystemResponse">
                <wsdl:part name="parameters"
element="tns:subscribePositioningSystemResponse"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="moveVehicleRequest">
                <wsdl:part name="parameters" element="tns:moveVehicle"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="moveVehicleResponse">
                <wsdl:part name="parameters" element="tns:moveVehicleResponse"></wsdl:part>
        </wsdl:message>
        <wsdl:portType name="C2Gis">
                <wsdl:operation name="getVideo"
                        sawsdl:modelReference="http://www.connect.com/ontology/media#ShowVideo">
                        <wsdl:output message="tns:getVideoRequest" />
                        <wsdl:input message="tns:getVideoResponse" />
                </wsdl:operation>
                <wsdl:operation name="panAndTilt"
                        sawsdl:modelReference="http://www.connect.com/ontology/media#PanTilt">
                        <wsdl:output message="tns:panAndTiltRequest"></wsdl:output>
                        <wsdl:input message="tns:panAndTiltResponse"></wsdl:input>
                </wsdl:operation>
                <wsdl:operation name="zoom"
                        sawsdl:modelReference="http://www.connect.com/ontology/media#Zoom">
                        <wsdl:output message="tns:zoomRequest"></wsdl:output>
                        <wsdl:input message="tns:zoomResponse"></wsdl:input>
                </wsdl:operation>
                <wsdl:operation name="getWeather"
                        sawsdl:modelReference="http://www.connect.com/ontology/media#GetWeather">
                        <wsdl:output message="tns:getWeatherRequest"></wsdl:output>
                        <wsdl:input message="tns:getWeatherResponse"></wsdl:input>
                </wsdl:operation>
                <wsdl:operation name="subscribePositioningSystem"
sawsdl:modelReference="http://www.connect.com/ontology/media#SubscribePositioning">
                        <wsdl:input message="tns:subscribePositioningSystemRequest"></wsdl:input>
                        <wsdl:output
message="tns:subscribePositioningSystemResponse"></wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="moveVehicle"
sawsdl:modelReference="http://www.connect.com/ontology/media#SetMovePattern">
                        <wsdl:input message="tns:moveVehicleRequest"></wsdl:input>
                        <wsdl:output message="tns:moveVehicleResponse"></wsdl:output>
                </wsdl:operation>
        </wsdl:portType>
        <wsdl:binding name="C2GisSOAP" type="tns:C2Gis">
                <soap:binding style="document"
                        transport="http://schemas.xmlsoap.org/soap/http" />
                <wsdl:operation name="getVideo">
                        <soap:operation
soapAction="http://services.connect.thalesgroup.com/getVideo" />
                        <wsdl:input>
                                <soap:body use="literal" />
                        </wsdl:input>
                        <wsdl:output>
                                <soap:body use="literal" />
                        </wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="panAndTilt">
```

```
                        <soap:operation
soapAction="http://services.connect.thalesgroup.com/panAndTilt" />
                        <wsdl:input>
                                <soap:body use="literal" />
                        </wsdl:input>
                        <wsdl:output>
                                <soap:body use="literal" />
                        </wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="zoom">
                        <soap:operation soapAction="http://services.connect.thalesgroup.com/zoom"
/>
                        <wsdl:input>
                                <soap:body use="literal" />
                        </wsdl:input>
                        <wsdl:output>
                                <soap:body use="literal" />
                        </wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="getWeather">
                        <soap:operation
soapAction="http://services.connect.thalesgroup.com/getWeather" />
                        <wsdl:input>
                                <soap:body use="literal" />
                        </wsdl:input>
                        <wsdl:output>
                                <soap:body use="literal" />
                        </wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="subscribePositioningSystem">
                        <soap:operation

        soapAction="http://services.connect.thalesgroup.com/subscribePositioningSystem" />
                        <wsdl:input>
                                <soap:body use="literal" />
                        </wsdl:input>
                        <wsdl:output>
                                <soap:body use="literal" />
                        </wsdl:output>
                </wsdl:operation>
        </wsdl:binding>
        <wsdl:service name="C2Gis">
                <wsdl:port binding="tns:C2GisSOAP" name="C2GisSOAP">
                        <soap:address location="http://www.example.org/" />
                </wsdl:port>
        </wsdl:service>
</wsdl:definitions>
```

7.8 NS 5 – Mobile Weather Station

7.8.1 SAWSDL

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
        xmlns:tns="http://services.connect.thalesgroup.com/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="WeatherStation"
        targetNamespace="http://services.connect.thalesgroup.com/"
xmlns:sawsdl="http://www.w3.org/2007/01/sawsdl#">
        <wsdl:types>
                <xsd:schema targetNamespace="http://services.connect.thalesgroup.com/">
                        <xsd:element name="put"
sawsdl:modelReference="http://www.connect.com/ontology/media#DataInfo">
                                <xsd:complexType>
                                        <xsd:sequence>
                                                <xsd:element name="info" type="tns:DataInfo" />
                                        </xsd:sequence>
                                </xsd:complexType>
                        </xsd:element>
                        <xsd:element name="putResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#Void">
                                <xsd:complexType>
                                        <xsd:sequence>
```

```
                                        </xsd:sequence>
                                </xsd:complexType>
                        </xsd:element>
                        <xsd:element name="get"
sawsdl:modelReference="http://www.connect.com/ontology/media#Void">
                                <xsd:complexType>
                                        <xsd:sequence>
                                        </xsd:sequence>
                                </xsd:complexType>
                        </xsd:element>
                        <xsd:element name="getResponse"
sawsdl:modelReference="http://www.connect.com/ontology/media#DataInfo">
                                <xsd:complexType>
                                        <xsd:sequence>

                                                <xsd:element name="out"
type="tns:DataInfoSet"></xsd:element>
                                        </xsd:sequence>
                                </xsd:complexType>
                        </xsd:element>

                        <xsd:element name="connectPrim">
                                <xsd:complexType>
                                        <xsd:sequence>
                                                <xsd:element name="connectInput" type="xsd:string"
minOccurs="0"></xsd:element>
                                        </xsd:sequence>
                                </xsd:complexType>
                        </xsd:element>
                        <xsd:element name="connectPrimResponse">
                                <xsd:complexType>
                                        <xsd:sequence>

                                                <xsd:element name="return" type="xsd:string"
minOccurs="0"></xsd:element>
                                        </xsd:sequence>
                                </xsd:complexType>
                        </xsd:element>

                        <xsd:complexType name="DataInfo">
                                <xsd:sequence>
                                        <xsd:element name="day" type="xsd:int"></xsd:element>
                                        <xsd:element name="month" type="xsd:int"></xsd:element>
                                        <xsd:element name="year" type="xsd:int"></xsd:element>
                                        <xsd:element name="hour" type="xsd:int"></xsd:element>
                                        <xsd:element name="minute" type="xsd:int"></xsd:element>
                                        <xsd:element name="temperature"
type="xsd:double"></xsd:element>
                                        <xsd:element name="humidity"
type="xsd:double"></xsd:element>
                                        <xsd:element name="dewpoint"
type="xsd:double"></xsd:element>
                                        <xsd:element name="barometer"
type="xsd:double"></xsd:element>
                                        <xsd:element name="windspeed"
type="xsd:double"></xsd:element>
                                        <xsd:element name="gutspeed"
type="xsd:double"></xsd:element>
                                        <xsd:element name="direction"
type="xsd:double"></xsd:element>
                                </xsd:sequence>
                        </xsd:complexType>

                        <xsd:complexType name="DataInfoSet">
                                <xsd:sequence>
                                        <xsd:element name="dataInfos" type="tns:DataInfo"
                                                minOccurs="0" maxOccurs="unbounded"></xsd:element>
                                </xsd:sequence>
                        </xsd:complexType>
                </xsd:schema>
        </wsdl:types>
        <wsdl:message name="putRequest">
                <wsdl:part element="tns:put" name="parameters" />
        </wsdl:message>
        <wsdl:message name="putResponse">
                <wsdl:part element="tns:putResponse" name="parameters" />
```

```
        </wsdl:message>
        <wsdl:message name="getRequest">
                <wsdl:part name="parameters" element="tns:get"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="getResponse">
                <wsdl:part name="parameters" element="tns:getResponse"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="connectPrimRequest">
                <wsdl:part name="parameters" element="tns:connectPrim"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="connectPrimResponse">
                <wsdl:part name="parameters" element="tns:connectPrimResponse"></wsdl:part>
        </wsdl:message>
        <wsdl:portType name="WeatherStation">
                <wsdl:operation name="put"
sawsdl:modelReference="http://www.connect.com/ontology/media#Put">
                        <wsdl:input message="tns:putRequest" />
                        <wsdl:output message="tns:putResponse" />
                </wsdl:operation>
                <wsdl:operation name="get"
sawsdl:modelReference="http://www.connect.com/ontology/media#Get">
                        <wsdl:input message="tns:getRequest"></wsdl:input>
                        <wsdl:output message="tns:getResponse"></wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="connectPrim">
                        <wsdl:input message="tns:connectPrimRequest"></wsdl:input>
                        <wsdl:output message="tns:connectPrimResponse"></wsdl:output>
                </wsdl:operation>
        </wsdl:portType>
        <wsdl:binding name="WeatherStationSOAP" type="tns:WeatherStation">
                <soap:binding style="document"
                        transport="http://schemas.xmlsoap.org/soap/http" />
                <wsdl:operation name="put">
                        <soap:operation soapAction="http://services.connect.thalesgroup.com/put"
/>
                        <wsdl:input>
                                <soap:body use="literal" />
                        </wsdl:input>
                        <wsdl:output>
                                <soap:body use="literal" />
                        </wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="get">
                        <soap:operation soapAction="http://services.connect.thalesgroup.com/get"
/>
                        <wsdl:input>
                                <soap:body use="literal" />
                        </wsdl:input>
                        <wsdl:output>
                                <soap:body use="literal" />
                        </wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="connectPrim">
                        <soap:operation
soapAction="http://services.connect.thalesgroup.com/connectPrim" />
                        <wsdl:input>
                                <soap:body use="literal" />
                        </wsdl:input>
                        <wsdl:output>
                                <soap:body use="literal" />
                        </wsdl:output>
                </wsdl:operation>
        </wsdl:binding>
        <wsdl:service name="WeatherStation">
                <wsdl:port binding="tns:WeatherStationSOAP" name="WeatherStationSOAP">
                        <soap:address location="http://www.example.org/" />
                </wsdl:port>
        </wsdl:service>
</wsdl:definitions>
```

7.9 NS 6 - Weather Service

7.9.1 SAWSDL

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
	xmlns:tns="http://services.connect.thalesgroup.com/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
	xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="WeatherService"
	targetNamespace="http://services.connect.thalesgroup.com/"
	xmlns:sawsdl="http://www.w3.org/2007/01/sawsdl#">
	<wsdl:types>
		<xsd:schema targetNamespace="http://services.connect.thalesgroup.com/">
			<xsd:element name="login"

	sawsdl:modelReference="http://www.connect.com/ontology/media#LoginParameters">
				<xsd:complexType>
					<xsd:sequence>
						<xsd:element name="login"
type="tns:AuthentificationData" />
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
			<xsd:element name="loginResponse"

	sawsdl:modelReference="http://www.connect.com/ontology/media#Token">
				<xsd:complexType>
					<xsd:sequence>
						<xsd:element name="out" type="tns:Token" />
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
			<xsd:element name="getWeather"

	sawsdl:modelReference="http://www.connect.com/ontology/media#ZipCode">
				<xsd:complexType>
					<xsd:sequence>
						<xsd:element name="zipCode"
type="xsd:string"></xsd:element>
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
			<xsd:element name="getWeatherResponse"

	sawsdl:modelReference="http://www.connect.com/ontology/media#WeatherData">
				<xsd:complexType>
					<xsd:sequence>
						<xsd:element name="out"
type="tns:WeatherData"></xsd:element>
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
			<xsd:element name="getWeatherWithCredentials"

	sawsdl:modelReference="http://www.connect.com/ontology/media#ZipCode
http://www.connect.com/ontology/media#Token">
				<xsd:complexType>
					<xsd:sequence>
						<xsd:element name="credential"
type="tns:Token"></xsd:element>
						<xsd:element name="zipCode"
type="xsd:string"></xsd:element>
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
			<xsd:element name="getWeatherWithCredentialsResponse"

	sawsdl:modelReference="http://www.connect.com/ontology/media#WeatherData">
				<xsd:complexType>
					<xsd:sequence>
						<xsd:element name="out"
type="tns:WeatherDataSet"></xsd:element>
					</xsd:sequence>
				</xsd:complexType>
```

```
                        </xsd:element>
                        <xsd:element name="connectPrim">
                                <xsd:complexType>
                                        <xsd:sequence>

                                                <xsd:element name="connectInput" type="xsd:string"
                                                        minOccurs="0"></xsd:element>
                                        </xsd:sequence>
                                </xsd:complexType>
                        </xsd:element>
                        <xsd:element name="connectPrimResponse">
                                <xsd:complexType>
                                        <xsd:sequence>

                                                <xsd:element name="return" type="xsd:string"
minOccurs="0"></xsd:element>
                                        </xsd:sequence>
                                </xsd:complexType>
                        </xsd:element>
                        <xsd:complexType name="AuthentificationData">
                                <xsd:sequence>
                                        <xsd:element name="login" type="xsd:string"></xsd:element>
                                        <xsd:element name="password"
type="xsd:string"></xsd:element>
                                </xsd:sequence>
                        </xsd:complexType>
                        <xsd:complexType name="Token">
                                <xsd:sequence>
                                        <xsd:element name="id" type="xsd:string"></xsd:element>
                                </xsd:sequence>
                        </xsd:complexType>

                        <xsd:complexType name="WeatherData"></xsd:complexType>

                        <xsd:complexType name="WeatherDataSet">
                                <xsd:sequence>
                                        <xsd:element name="datas" type="tns:WeatherData"
                                                minOccurs="0" maxOccurs="unbounded"></xsd:element>
                                </xsd:sequence>
                        </xsd:complexType>
                </xsd:schema>
        </wsdl:types>
        <wsdl:message name="loginRequest">
                <wsdl:part element="tns:login" name="parameters" />
        </wsdl:message>
        <wsdl:message name="loginResponse">
                <wsdl:part element="tns:loginResponse" name="parameters" />
        </wsdl:message>
        <wsdl:message name="getWeatherRequest">
                <wsdl:part name="parameters" element="tns:getWeather"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="getWeatherResponse">
                <wsdl:part name="parameters" element="tns:getWeatherResponse"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="getWeatherWithCredentialsRequest">
                <wsdl:part name="parameters"
element="tns:getWeatherWithCredentials"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="getWeatherWithCredentialsResponse">
                <wsdl:part name="parameters"
element="tns:getWeatherWithCredentialsResponse"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="connectPrimRequest">
                <wsdl:part name="parameters" element="tns:connectPrim"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="connectPrimResponse">
                <wsdl:part name="parameters" element="tns:connectPrimResponse"></wsdl:part>
        </wsdl:message>
        <wsdl:portType name="WeatherService">
                <wsdl:operation name="login"
                        sawsdl:modelReference="http://www.connect.com/ontology/media#Login">
                        <wsdl:input message="tns:loginRequest" />
                        <wsdl:output message="tns:loginResponse" />
                </wsdl:operation>
                <wsdl:operation name="getWeather"
                        sawsdl:modelReference="http://www.connect.com/ontology/media#GetWeather">
```

```
                        <wsdl:input message="tns:getWeatherRequest"></wsdl:input>
                        <wsdl:output message="tns:getWeatherResponse"></wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="getWeatherWithCredentials"

        sawsdl:modelReference="http://www.connect.com/ontology/media#GetWeatherWithCredentials">
                        <wsdl:input message="tns:getWeatherWithCredentialsRequest"></wsdl:input>
                        <wsdl:output
message="tns:getWeatherWithCredentialsResponse"></wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="connectPrim">
                        <wsdl:input message="tns:connectPrimRequest"></wsdl:input>
                        <wsdl:output message="tns:connectPrimResponse"></wsdl:output>
                </wsdl:operation>
        </wsdl:portType>
        <wsdl:binding name="WeatherServiceSOAP" type="tns:WeatherService">
                <soap:binding style="document"
                        transport="http://schemas.xmlsoap.org/soap/http" />
                <wsdl:operation name="login">
                        <soap:operation
soapAction="http://services.connect.thalesgroup.com/login" />
                        <wsdl:input>
                                <soap:body use="literal" />
                        </wsdl:input>
                        <wsdl:output>
                                <soap:body use="literal" />
                        </wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="getWeather">
                        <soap:operation
soapAction="http://services.connect.thalesgroup.com/getWeather" />
                        <wsdl:input>
                                <soap:body use="literal" />
                        </wsdl:input>
                        <wsdl:output>
                                <soap:body use="literal" />
                        </wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="getWeatherWithCredentials">
                        <soap:operation

        soapAction="http://services.connect.thalesgroup.com/getWeatherWithCredentials" />
                        <wsdl:input>
                                <soap:body use="literal" />
                        </wsdl:input>
                        <wsdl:output>
                                <soap:body use="literal" />
                        </wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="connectPrim">
                        <soap:operation
soapAction="http://services.connect.thalesgroup.com/connectPrim" />
                        <wsdl:input>
                                <soap:body use="literal" />
                        </wsdl:input>
                        <wsdl:output>
                                <soap:body use="literal" />
                        </wsdl:output>
                </wsdl:operation>
        </wsdl:binding>
        <wsdl:service name="WeatherService">
                <wsdl:port binding="tns:WeatherServiceSOAP" name="WeatherServiceSOAP">
                        <soap:address location="http://www.example.org/" />
                </wsdl:port>
        </wsdl:service>
</wsdl:definitions>
```

7.10 NS 7.1 - Positioning - Country A

7.10.1 SAWSDL

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
	xmlns:tns="http://services.connect.thalesgroup.com/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
	xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="PositioningDDSSystem"
	targetNamespace="http://services.connect.thalesgroup.com/"
	xmlns:sawsdl="http://www.w3.org/2007/01/sawsdl#">
	<wsdl:types>
		<xsd:schema targetNamespace="http://services.connect.thalesgroup.com/">
			<xsd:element name="subscribe"

	sawsdl:modelReference="http://www.connect.com/ontology/media#Void">
				<xsd:complexType>
					<xsd:sequence>
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
			<xsd:element name="subscribeResponse"

	sawsdl:modelReference="http://www.connect.com/ontology/media#Void">
				<xsd:complexType>
					<xsd:sequence>
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
			<xsd:element name="publish"

	sawsdl:modelReference="http://www.connect.com/ontology/media#Position">
				<xsd:complexType>
					<xsd:sequence>
						<xsd:element name="PositionComplex">
							<xsd:complexType>
								<xsd:sequence>
									<xsd:element name="id"
type="xsd:string" />
									<xsd:element name="latitude"
type="xsd:double" />
									<xsd:element
name="longitude" type="xsd:double" />
									<xsd:element
name="timestamp" type="xsd:double" />
								</xsd:sequence>
							</xsd:complexType>
						</xsd:element>
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
			<xsd:element name="publishResponse"

	sawsdl:modelReference="http://www.connect.com/ontology/media#Void">
				<xsd:complexType>
					<xsd:sequence>
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
		</xsd:schema>
	</wsdl:types>
	<wsdl:message name="subscribeRequest">
		<wsdl:part element="tns:subscribe" name="parameters" />
	</wsdl:message>
	<wsdl:message name="subscribeResponse">
		<wsdl:part element="tns:subscribeResponse" name="parameters" />
	</wsdl:message>
	<wsdl:message name="publishRequest">
		<wsdl:part name="parameters" element="tns:publish"></wsdl:part>
	</wsdl:message>
	<wsdl:message name="publishResponse">
		<wsdl:part name="parameters" element="tns:publishResponse"></wsdl:part>
	</wsdl:message>
	<wsdl:message name="registerRequest">
		<wsdl:part name="parameters" element="tns:register"></wsdl:part>
	</wsdl:message>
	<wsdl:message name="registerResponse">
		<wsdl:part name="parameters" element="tns:registerResponse"></wsdl:part>
	</wsdl:message>
	<wsdl:portType name="PositioningDDSSystem">
```

```
                <wsdl:operation name="subscribe"

        sawsdl:modelReference="http://www.connect.com/ontology/media#SubscribePositioning">
                        <wsdl:input message="tns:subscribeRequest" />
                        <wsdl:output message="tns:subscribeResponse" />
                </wsdl:operation>
                <wsdl:operation name="publish"

        sawsdl:modelReference="http://www.connect.com/ontology/media#PublishPosition">
                        <wsdl:input message="tns:publishRequest"></wsdl:input>
                        <wsdl:output message="tns:publishResponse"></wsdl:output>
                </wsdl:operation>
        </wsdl:portType>
        <wsdl:binding name="PositioningDDSSystemSOAP" type="tns:PositioningDDSSystem">
                <soap:binding style="document"
                        transport="http://schemas.xmlsoap.org/soap/http" />
                <wsdl:operation name="subscribe">
                        <soap:operation
soapAction="http://services.connect.thalesgroup.com/subscribe" />
                        <wsdl:input>
                                <soap:body use="literal" />
                        </wsdl:input>
                        <wsdl:output>
                                <soap:body use="literal" />
                        </wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="publish">
                        <soap:operation
soapAction="http://services.connect.thalesgroup.com/publish" />
                        <wsdl:input>
                                <soap:body use="literal" />
                        </wsdl:input>
                        <wsdl:output>
                                <soap:body use="literal" />
                        </wsdl:output>
                </wsdl:operation>
        </wsdl:binding>
        <wsdl:service name="PositioningDDSSystem">
                <wsdl:port binding="tns:PositioningDDSSystemSOAP"
name="PositioningDDSSystemSOAP">
                        <soap:address location="http://www.example.org/" />
                </wsdl:port>
        </wsdl:service>
</wsdl:definitions>
```

7.11 NS 7.2 - Positioning - Country B

7.11.1 SAWSDL

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
        xmlns:tns="http://services.connect.thalesgroup.com/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="PositioningJMSSystem"
        targetNamespace="http://services.connect.thalesgroup.com/"
        xmlns:sawsdl="http://www.w3.org/2007/01/sawsdl#">
        <wsdl:types>
                <xsd:schema targetNamespace="http://services.connect.thalesgroup.com/">
                        <xsd:element name="subscribe"

        sawsdl:modelReference="http://www.connect.com/ontology/media#Void">
                                <xsd:complexType>
                                        <xsd:sequence>
                                        </xsd:sequence>
                                </xsd:complexType>
                        </xsd:element>
                        <xsd:element name="subscribeResponse"

        sawsdl:modelReference="http://www.connect.com/ontology/media#Void">
                                <xsd:complexType>
                                        <xsd:sequence>
                                        </xsd:sequence>
                                </xsd:complexType>
```

```
                        </xsd:element>
                        <xsd:element name="publish"

        sawsdl:modelReference="http://www.connect.com/ontology/media#PositionString">
                                <xsd:complexType>
                                        <xsd:sequence>
                                                <xsd:element name="Position" type="xsd:string" />
                                        </xsd:sequence>
                                </xsd:complexType>
                        </xsd:element>
                        <xsd:element name="publishResponse"

        sawsdl:modelReference="http://www.connect.com/ontology/media#Void">
                                <xsd:complexType>
                                        <xsd:sequence>
                                        </xsd:sequence>
                                </xsd:complexType>
                        </xsd:element>
                </xsd:schema>
        </wsdl:types>
        <wsdl:message name="subscribeRequest">
                <wsdl:part element="tns:subscribe" name="parameters" />
        </wsdl:message>
        <wsdl:message name="subscribeResponse">
                <wsdl:part element="tns:subscribeResponse" name="parameters" />

        </wsdl:message>
        <wsdl:message name="publishRequest">
                <wsdl:part name="parameters" element="tns:publish"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="publishResponse">
                <wsdl:part name="parameters" element="tns:publishResponse"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="registerRequest">
                <wsdl:part name="parameters" element="tns:register"></wsdl:part>
        </wsdl:message>
        <wsdl:message name="registerResponse">
                <wsdl:part name="parameters" element="tns:registerResponse"></wsdl:part>
        </wsdl:message>
        <wsdl:portType name="PositioningJMSSystem">
                <wsdl:operation name="subscribe"

        sawsdl:modelReference="http://www.connect.com/ontology/media#SubscribePositioning">
                        <wsdl:input message="tns:subscribeRequest" />
                        <wsdl:output message="tns:subscribeResponse" />
                </wsdl:operation>
                <wsdl:operation name="publish"

        sawsdl:modelReference="http://www.connect.com/ontology/media#PublishPosition">
                        <wsdl:input message="tns:publishRequest"></wsdl:input>
                        <wsdl:output message="tns:publishResponse"></wsdl:output>
                </wsdl:operation>
        </wsdl:portType>
        <wsdl:binding name="PositioningJMSSystemSOAP" type="tns:PositioningJMSSystem">
                <soap:binding style="document"
                        transport="http://schemas.xmlsoap.org/soap/http" />
                <wsdl:operation name="subscribe">
                        <soap:operation
soapAction="http://services.connect.thalesgroup.com/subscribe" />
                        <wsdl:input>
                                <soap:body use="literal" />
                        </wsdl:input>
                        <wsdl:output>
                                <soap:body use="literal" />
                        </wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="publish">
                        <soap:operation
soapAction="http://services.connect.thalesgroup.com/publish" />
                        <wsdl:input>
                                <soap:body use="literal" />
                        </wsdl:input>
                        <wsdl:output>
                                <soap:body use="literal" />
                        </wsdl:output>
                </wsdl:operation>
```

```
        </wsdl:binding>
        <wsdl:service name="PositioningJMSSystem">
                <wsdl:port binding="tns:PositioningJMSSystemSOAP"
name="PositioningJMSSystemSOAP">
                        <soap:address location="http://www.example.org/" />
                </wsdl:port>
        </wsdl:service>
</wsdl:definitions>
```

7.12 GMES-related Ontology

```
<?xml version="1.0"?>


<!DOCTYPE Ontology [
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>


<Ontology xmlns="http://www.w3.org/2002/07/owl#"
     xml:base="http://www.connect.com/ontology/media"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"
     ontologyIRI="http://www.connect.com/ontology/media">
    <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>
    <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
    <Prefix name="" IRI="http://www.w3.org/2002/07/owl#"/>
    <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
    <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>
    <Declaration>
        <Class IRI="#AVIFormat"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Acknowledgement"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Affordances"/>
    </Declaration>
    <Declaration>
        <Class IRI="#CIRCLE"/>
    </Declaration>
    <Declaration>
        <Class IRI="#DIAMOND"/>
    </Declaration>
    <Declaration>
        <Class IRI="#ErrorMessage"/>
    </Declaration>
    <Declaration>
        <Class IRI="#FixedVideoSource"/>
    </Declaration>
    <Declaration>
        <Class IRI="#FlyingMachine"/>
    </Declaration>
    <Declaration>
        <Class IRI="#FlyingVideoSource"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Get"/>
    </Declaration>
    <Declaration>
        <Class IRI="#GetRTSPAddress"/>
    </Declaration>
    <Declaration>
        <Class IRI="#GetWeather"/>
    </Declaration>
    <Declaration>
        <Class IRI="#GetWeatherWithCredentials"/>
    </Declaration>
    <Declaration>
```

```
        <Class IRI="#Login"/>
    </Declaration>
    <Declaration>
        <Class IRI="#LoginParameters"/>
    </Declaration>
    <Declaration>
        <Class IRI="#MJPEGAddress"/>
    </Declaration>
    <Declaration>
        <Class IRI="#MP4Format"/>
    </Declaration>
    <Declaration>
        <Class IRI="#MobileVideoSource"/>
    </Declaration>
    <Declaration>
        <Class IRI="#MovePattern"/>
    </Declaration>
    <Declaration>
        <Class IRI="#MoveState"/>
    </Declaration>
    <Declaration>
        <Class IRI="#PanTilt"/>
    </Declaration>
    <Declaration>
        <Class IRI="#PanTiltParameters"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Parameters"/>
    </Declaration>
    <Declaration>
        <Class IRI="#PatrolVideoSource"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Position"/>
    </Declaration>
    <Declaration>
        <Class IRI="#PositionComplex"/>
    </Declaration>
    <Declaration>
        <Class IRI="#PositionString"/>
    </Declaration>
    <Declaration>
        <Class IRI="#PositioningSource"/>
    </Declaration>
    <Declaration>
        <Class IRI="#PositioningSourceDDS"/>
    </Declaration>
    <Declaration>
        <Class IRI="#PositioningSourceJMS"/>
    </Declaration>
    <Declaration>
        <Class IRI="#PublishPosition"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Put"/>
    </Declaration>
    <Declaration>
        <Class IRI="#RTSPAddress"/>
    </Declaration>
    <Declaration>
        <Class IRI="#RecordVideo"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Rotate"/>
    </Declaration>
    <Declaration>
        <Class IRI="#SQUARE"/>
    </Declaration>
    <Declaration>
        <Class IRI="#ServicesOperations"/>
    </Declaration>
    <Declaration>
        <Class IRI="#SetMovePattern"/>
    </Declaration>
    <Declaration>
        <Class IRI="#StartPatrol"/>
```

```
    </Declaration>
    <Declaration>
        <Class IRI="#StopPatrol"/>
    </Declaration>
    <Declaration>
        <Class IRI="#SubscribePositioning"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Token"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVAccessToken"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVAccessTokenString"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVAccessTokenWithStreamId"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVAccessTokenWithTime"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVAirborneOperations"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVAuthargs"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVAuthenticatedOperations"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVAuthorizeStream"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVClientsList"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVCoordinates"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVGetCoordinates"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVGetStreamClients"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVGetStreamDescriptions"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVLand"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVLogin"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVLogout"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVMoveBack"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVMoveDown"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVMoveForward"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVMoveLeft"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVMoveRight"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVMoveUp"/>
    </Declaration>
```

```
    <Declaration>
        <Class IRI="#UAVOperations"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVRequest"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVResponse"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVStreamDescriptionList"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVStreamUrl"/>
    </Declaration>
    <Declaration>
        <Class IRI="#UAVTakeoff"/>
    </Declaration>
    <Declaration>
        <Class IRI="#URL"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Vehicle"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Video"/>
    </Declaration>
    <Declaration>
        <Class IRI="#VideoDuration"/>
    </Declaration>
    <Declaration>
        <Class IRI="#VideoParameters"/>
    </Declaration>
    <Declaration>
        <Class IRI="#VideoResolution"/>
    </Declaration>
    <Declaration>
        <Class IRI="#VideoSource"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Void"/>
    </Declaration>
    <Declaration>
        <Class IRI="#WeatherData"/>
    </Declaration>
    <Declaration>
        <Class IRI="#WeatherInfo"/>
    </Declaration>
    <Declaration>
        <Class IRI="#WeatherStory"/>
    </Declaration>
    <Declaration>
        <Class IRI="#WeatherWithForcast"/>
    </Declaration>
    <Declaration>
        <Class IRI="#ZipCode"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Zoom"/>
    </Declaration>
    <Declaration>
        <Class IRI="#ZoomFactor"/>
    </Declaration>
    <EquivalentClasses>
        <Class IRI="#GetRTSPAddress"/>
        <Class IRI="#UAVAuthorizeStream"/>
    </EquivalentClasses>
    <SubClassOf>
        <Class IRI="#AVIFormat"/>
        <Class IRI="#Video"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Acknowledgement"/>
        <Class IRI="#Parameters"/>
    </SubClassOf>
    <SubClassOf>
```

```
        <Class IRI="#CIRCLE"/>
        <Class IRI="#MovePattern"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#DIAMOND"/>
        <Class IRI="#MovePattern"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#ErrorMessage"/>
        <Class IRI="#Parameters"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#FixedVideoSource"/>
        <Class IRI="#VideoSource"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#FlyingMachine"/>
        <Class IRI="#Vehicle"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#FlyingVideoSource"/>
        <Class IRI="#MobileVideoSource"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Get"/>
        <Class IRI="#ServicesOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#GetRTSPAddress"/>
        <Class IRI="#ServicesOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#GetWeather"/>
        <Class IRI="#ServicesOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#GetWeatherWithCredentials"/>
        <Class IRI="#ServicesOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Login"/>
        <Class IRI="#ServicesOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#LoginParameters"/>
        <Class IRI="#Parameters"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#MJPEGAddress"/>
        <Class IRI="#URL"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#MP4Format"/>
        <Class IRI="#Video"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#MobileVideoSource"/>
        <Class IRI="#VideoSource"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#MovePattern"/>
        <Class IRI="#Parameters"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#MoveState"/>
        <Class IRI="#Parameters"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#PanTilt"/>
        <Class IRI="#ServicesOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#PanTiltParameters"/>
        <Class IRI="#Parameters"/>
    </SubClassOf>
    <SubClassOf>
```

```
        <Class IRI="#PatrolVideoSource"/>
        <Class IRI="#VideoSource"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Position"/>
        <Class IRI="#Parameters"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#PositionComplex"/>
        <Class IRI="#Position"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#PositionString"/>
        <Class IRI="#Position"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#PositioningSource"/>
        <Class IRI="#Affordances"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#PositioningSourceDDS"/>
        <Class IRI="#PositioningSource"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#PositioningSourceJMS"/>
        <Class IRI="#PositioningSource"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#PublishPosition"/>
        <Class IRI="#ServicesOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Put"/>
        <Class IRI="#ServicesOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#RTSPAddress"/>
        <Class IRI="#URL"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#RecordVideo"/>
        <Class IRI="#ServicesOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Rotate"/>
        <Class IRI="#ServicesOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#SQUARE"/>
        <Class IRI="#MovePattern"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#SetMovePattern"/>
        <Class IRI="#ServicesOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#StartPatrol"/>
        <Class IRI="#ServicesOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#StopPatrol"/>
        <Class IRI="#ServicesOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#SubscribePositioning"/>
        <Class IRI="#ServicesOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Token"/>
        <Class IRI="#Parameters"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVAccessToken"/>
        <Class IRI="#Token"/>
    </SubClassOf>
    <SubClassOf>
```

```
        <Class IRI="#UAVAccessToken"/>
        <Class IRI="#UAVRequest"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVAccessTokenString"/>
        <Class IRI="#Token"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVAccessTokenString"/>
        <Class IRI="#UAVResponse"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVAccessTokenWithStreamId"/>
        <Class IRI="#UAVRequest"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVAccessTokenWithTime"/>
        <Class IRI="#UAVRequest"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVAirborneOperations"/>
        <Class IRI="#UAVAuthenticatedOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVAuthargs"/>
        <Class IRI="#LoginParameters"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVAuthenticatedOperations"/>
        <Class IRI="#UAVOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVAuthorizeStream"/>
        <Class IRI="#UAVAuthenticatedOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVClientsList"/>
        <Class IRI="#UAVResponse"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVCoordinates"/>
        <Class IRI="#UAVResponse"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVGetCoordinates"/>
        <Class IRI="#UAVAuthenticatedOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVGetStreamClients"/>
        <Class IRI="#UAVAuthenticatedOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVGetStreamDescriptions"/>
        <Class IRI="#UAVAuthenticatedOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVLand"/>
        <Class IRI="#UAVAirborneOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVLogin"/>
        <Class IRI="#Login"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVLogin"/>
        <Class IRI="#UAVOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVLogout"/>
        <Class IRI="#UAVAuthenticatedOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVMoveBack"/>
        <Class IRI="#UAVAirborneOperations"/>
    </SubClassOf>
    <SubClassOf>
```

```
        <Class IRI="#UAVMoveDown"/>
        <Class IRI="#UAVAirborneOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVMoveForward"/>
        <Class IRI="#UAVAirborneOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVMoveLeft"/>
        <Class IRI="#UAVAirborneOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVMoveRight"/>
        <Class IRI="#UAVAirborneOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVMoveUp"/>
        <Class IRI="#UAVAirborneOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVOperations"/>
        <Class IRI="#ServicesOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVRequest"/>
        <Class IRI="#Parameters"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVResponse"/>
        <Class IRI="#Parameters"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVStreamDescriptionList"/>
        <Class IRI="#UAVResponse"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVStreamUrl"/>
        <Class IRI="#UAVResponse"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#UAVTakeoff"/>
        <Class IRI="#UAVAuthenticatedOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#URL"/>
        <Class IRI="#Parameters"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Vehicle"/>
        <Class IRI="#Affordances"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Video"/>
        <Class IRI="#Parameters"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#VideoDuration"/>
        <Class IRI="#Parameters"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#VideoParameters"/>
        <Class IRI="#Parameters"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#VideoResolution"/>
        <Class IRI="#Parameters"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#VideoSource"/>
        <Class IRI="#Affordances"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Void"/>
        <Class IRI="#Parameters"/>
    </SubClassOf>
    <SubClassOf>
```

```
        <Class IRI="#WeatherData"/>
        <Class IRI="#Parameters"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#WeatherInfo"/>
        <Class IRI="#Affordances"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#WeatherStory"/>
        <Class IRI="#WeatherInfo"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#WeatherWithForcast"/>
        <Class IRI="#WeatherInfo"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#ZipCode"/>
        <Class IRI="#Parameters"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Zoom"/>
        <Class IRI="#ServicesOperations"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#ZoomFactor"/>
        <Class IRI="#Parameters"/>
    </SubClassOf>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#Affordances</IRI>
        <Literal>Simple concept for categorization purpose.</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#FlyingMachine</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">This represents a flying machine; like a
UAV</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#FlyingVideoSource</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">This represents a flying video source, like a
camera attached to a UAV</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#Parameters</IRI>
        <Literal>Simple concept for categorization purpose.</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#PositionComplex</IRI>
        <Literal>represents a structured position</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#PositionString</IRI>
        <Literal>represents a simple string delemited by separators</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#ServicesOperations</IRI>
        <Literal>Simple concept for categorization purpose.</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#UAVAirborneOperations</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Operations possible only while the drone is
airborne</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#UAVAuthargs</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">The username and password needed for logging
into a UAV</Literal>
    </AnnotationAssertion>
```

```
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#UAVAuthenticatedOperations</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Operations possible only after getting an
access token from the UAV</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#UAVAuthorizeStream</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Very similar to the GetRTSPAddress, but in
addition to getting the address, it also authorized the client to connect to that address. If
not, the (UAV) camera will reject requests for actual videos.</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#UAVGetCoordinates</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Get the (X,Y,Z,Roll, Pitch, Yaw) coordinates
of the UAV</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#UAVGetStreamClients</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Get the list of clients connected to a
particular video source on the UAV.</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#UAVGetStreamDescriptions</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Get the list of stream source descriptions
available on this UAV&amp;#39;s camera.</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#UAVLand</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Land the UAV</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#UAVLogin</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">login and get an access token for UAV
operations</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#UAVLogout</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Invalidates the access token.</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#UAVMoveBack</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Move the UAV back for some
milliseconds.</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#UAVMoveDown</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Move the UAV down for some
milliseconds.</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#UAVMoveForward</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Move the UAV forward for some
milliseconds.</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#UAVMoveLeft</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Move the UAV left for some
milliseconds.</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#UAVMoveRight</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Move the UAV right for some
milliseconds.</Literal>
```

```
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#UAVMoveUp</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Move the UAV up for some
milliseconds.</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#UAVOperations</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Operations relevant to the UAV</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#UAVTakeoff</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Causes the UAV to get airborne.</Literal>
    </AnnotationAssertion>
</Ontology>
<!-- Generated by the OWL API (version 3.0.0.1413) http://owlapi.sourceforge.net -->
```