**RECONSURVE (A Reconfigurable Surveillance System with Smart Sensors and Communication)**

**Review of the State of the Art in**

**Maritime Surveillance Systems**

20 May 2015

Contents

[1 Table of Figures and Tables 3](#_Toc419894961)

[2 Executive Summary 4](#_Toc419894962)

[3 Introduction 6](#_Toc419894963)

[4 State-of-the-Art in Technology 10](#_Toc419894964)

[4.1 Semantic Interoperability 10](#_Toc419894965)

[4.2 Sensor Data Processing 12](#_Toc419894966)

[4.3 Behavioral Analysis 14](#_Toc419894967)

[4.4 State-of-the-Art in Communication and Networks 17](#_Toc419894968)

[4.5 Intelligent Interfaces 19](#_Toc419894969)

[5 State of the Art in Standards 22](#_Toc419894970)

[5.1 OGC Sensor Web Enablement 22](#_Toc419894971)

[5.2 Description Logics Reasoners 23](#_Toc419894972)

[5.3 Joint Consultation, Command and Control Information Exchange Data Model (JC3IEDM) 24](#_Toc419894973)

[5.4 Semantic Web Technologies 25](#_Toc419894974)

[5.4.1 Semantic Web 25](#_Toc419894975)

[5.4.2 Ontology 27](#_Toc419894976)

[5.4.3 Web Ontology Language (OWL) 28](#_Toc419894977)

[5.5 Web Services 29](#_Toc419894978)

[5.6 Data Distribution Services (DDS) 34](#_Toc419894979)

[5.7 UN/CEFACT Unqualified Data Types 35](#_Toc419894980)

[5.8 OASIS Common Alerting Protocol (CAP) 37](#_Toc419894981)

[6 Conclusion 38](#_Toc419894982)

# Table of Figures and Tables

[Figure 1 A typical surveillance system stack 5](#_Toc419897546)

[Figure 1 System Architecture 7](#_Toc419897547)

[Figure 3 Images from RECONSURVE’s civilian ship image dataset 13](#_Toc419897548)

[Figure 4 The structure of Data Distribution Service 35](#_Toc419897549)

[Table 1 Unqualified Data Types 36](#_Toc419897576)

# Executive Summary

The RECONSURVE (A Reconfigurable Surveillance System with Smart Sensors and Communication) project, which started at the beginning of 2011, was motivated by and aimed to address the need to control the rapidly increasing number and complexity of maritime surveillance issues such as illegal immigration especially using small vessels, interoperability between heterogeneous systems, automated cost-effective and efficient decision support. This document provides an update to state-of-the-art as of mid-2015, at the end of the project.

In a broad perspective, technical components of a typical surveillance system comprise a set of sensors, and systems that manages the sensors and processes the information coming from the sensors. Depending on the needs, the system may contain EO/IR cameras, radars, acoustic sensors, seismic sensors, etc. The sensors may function individually or be put together in the form of sensor networks. The data received from the sensors is processed using signal and image processing algorithms, and subsequently converted into a format suitable for displaying to a human operator or sent to another system for automated processing that may fuse data coming from multiple sensors. The algorithms further perform behavioural analysis and/or provide intelligence to prevent information overload and the limit the information to the essential need for safety and security. The results can be then presented to the operator creating situational-awareness and visualization software in a form that can be understood by the operator and acted on. The information can be stored into an information repository conforming to the Resource Description Framework (RDF) standard. This enables regular querying for pattern recognition and could in turn activate further triggers and alarms. These triggers and alarms are presented on operator screens or are translated into concrete actions – such as messages to the appropriate agencies. A surveillance system may also have a wired or wireless communication interface that sends and receives messages to and from other external systems, as illustrated in Figure 1.

Figure 1 A typical surveillance system stack

Many surveillance systems that fit into this general structure were built and deployed over the last decade around the world. The state-of-the-art for each layer in this system stack can be characterized as follows:

* Modern surveillance system user interfaces often are characterized by complex operator interfaces centered on multiple display visualization capabilities. There is a trend in the industry to advance operator interfaces to equip with innovative visualization components. This component will comprise a unified display as a central part of the operator interface but also addresses the need for other operator aids such as a voice interface for hands-free queries in conjunction with visual and tactile interfaces. Automatic dissemination of critical event notifications or warnings using the so-called “Smart-Push” communication is also under consideration.
* The case is also similar for sensors feeding observation data to the maritime surveillance system: it is too much and overloads operators. The industry aims to facilitate and complement the operator’s analysis tasks and to reduce the workload by analyzing the sensor data, verifying it by means of additional sensor modalities, performing high semantic-level reasoning to detect suspicious situations, and initiating actions such as alarming and sending comprehensive messages.
* The current surveillance systems function as static and standalone systems. There is, however, consensus that surveillance systems should not be limited anymore to static environments and scenarios, because threats and the scenarios change as soon as the environments become more protected. Furthermore, a great benefit is foreseen if two or more surveillance systems, especially if they are geographically in close proximity, can exchange information and share situational awareness. Ideally, surveillance systems in the future should operate as systems of systems consisting of existing legacy systems as well as entirely new systems. This capability should be integrated in the design of the systems in terms of flexibility, re-targeting the use of the system, which requires significant integration efforts and system re-design experiments.
* Interoperability of surveillance systems to allow fluent metadata conversion and open formats for logging, alerting and smooth communication between different surveillance systems is also desired. At lower levels of granularity, at device level, the sensors can be plug and play components of a surveillance system. At higher levels of granularity, the surveillance systems can also plug and play into each other, as systems of systems. They will be able to exchange and fuse information, especially if they are in close proximity, share situational awareness and collaborate. Achieving plug and play interoperability involves not a single standard but a collection of standards addressing different layers in the interoperability stack. However, there are several alternative standards to be chosen from for each layer and some standards specify a range of standards for a layer.

RECONSURVE project contributed to advancing the state-of-the-art primarily in behavioral analysis and image and signal processing layers. This report presents the state of the art in relation to maritime surveillance systems and standards as of the end of the project.

# Introduction

This report presents the state of the art in relation to the outcome of the RECONSURVE project, specifically in maritime surveillance systems and standards. The RECONSURVE project started with the following architecture:

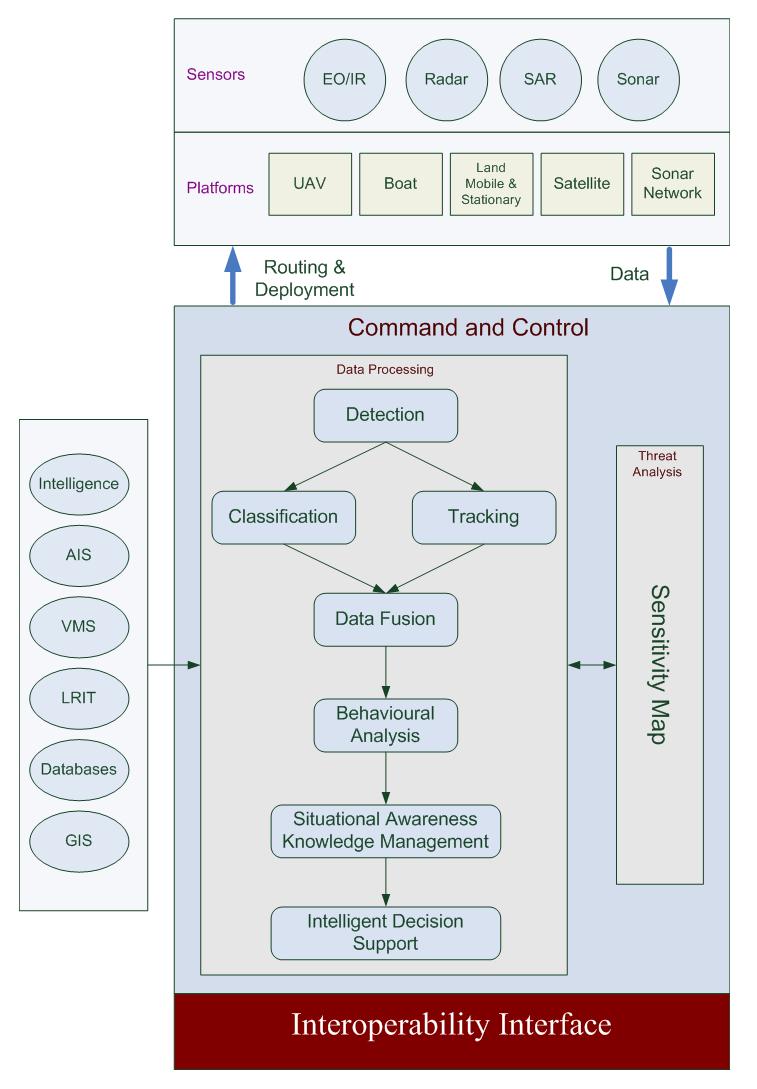


Figure 1 System Architecture

At the start of the project, although there were some maritime surveillance systems available, they lacked the technical and architectural maturity to tackle all these requirements at once. Some companies had some of the RECONSURVE subsystems as individual, disparate systems; some had “unified” systems that display several data feeds all at once without the critical automated decision making and support component and yet some had an integrated system with only very limited algorithmic capabilities.

The RECONSURVE project advanced the state-of-the-art in maritime surveillance systems with the integration of the following components:

* UAV capabilities
* Sonar network capabilities
* Interoperability
* Automatic detection and classification
* Multi-Sensor Data Analysis

The project achieved the following major outcomes:

* Interoperability.
* Small vessel detection & classification capability.
* Cost effectiveness in a wide-area sea border surveillance system.

This deliverable presents the current state of the art in the technologies and standards in relation to the scope of RECONSURVE Project.

The list of European projects that are conducted about the same timeline with the RECONSURVE project is illustrated in the following table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Name** | **Cooperative Programme** | **Time period** | **Technical Focus** | **Differences with the RECONSURVE project** |
| SPY | ITEA2 | 2011-2013 | An automated, intelligent surveillance and rescue framework adapted to the mobile environment by means of the use of wireless infrastructures | Primarily focused on urban surveillance with smart sensor processing, but did not aim to achieve semantic interoperability capabilities. |
| PERSEUS | FP7 Security | 2011-2014 | EU maritime surveillance system demonstration integrating existing national and communitarian installations | Related to syntactical interoperability of surveillance applications as it defines a common data model and the external applications interfaces but it lacked semantic and workflow definitions |
| SEABILLA | FP7 Security |  | Integration of legacy maritime surveillance systems and, improvement in shareable common maritime picture targeted to detect non cooperative boats and illicit activities | Related to integration of legacy maritime surveillance systems and situational awareness functionalities but different in not addressing semantic interoperability of surveillance systems for future systems and smart surveillance functionalities for different domains. |
| INDIGO | FP7 | 2010-2013 | 3D interactive and realistic visualization of the crisis environment, including data coming from the field, simulation results, and building interiors. | Focusing on visualization, but do not provide voice interface to the operator. |
| ViCoMo | ITEA2 | 2009-2012 | Advanced video-interpretation algorithms to enhance images acquired with multiple camera systems and by modelling the context in which such systems are used, | Although multi-camera analysis is addressed, it does not address back-tracing of events in large (>2000) camera systems. |
| FEDSS | ITEA2 | 2013-2016 | Maritime Surveillance, semantic information | FEDSS is related to data fusion from several sensors. Experience in image analysis and suspicious pattern detection could be applied to RECONSURVE. |
| SAFECITY | FP7 | 2011-2013 | Urban surveillance. Smart Public safety and security in cities | SAFECITY is more centred in solving future internet and Internet of the Things issues like M2M communications, and is a Use Case of FI-WARE. Aspects related to interoperability and Smart Sensors and the general Architecture could be studied in maritime surveillance |

# State-of-the-Art in Technology

## Semantic Interoperability

Achieving plug and play solutions is a common goal in the engineering world. This goal has been pursued and achieved successfully in numerous market segments for countless product categories at different levels of granularity. A USB flash memory is an example of a plug and play solution at the device level. Internet is an example at the systems of systems level. Today, plug & play capabilities do not exist in the market for surveillance systems. Therefore, we review the state-of-the-art from the perspective of the underlying technological building blocks required to build the mentioned capability at the different levels of the surveillance system stack. Syntactic and functional interoperability between the surveillance systems and the sensors are the necessary condition for further enabling semantic interoperability. Syntactic and functional interoperability is required in order to define standardized interfaces. Indeed, standards are the cornerstones of the interoperability in this regard.

Providing interoperability across different organizations requires a robust yet agile information model in order to correctly select and use services and exchange information. In surveillance systems the Joint Consultation, Command, and Control Information Exchange Data Model (JC3IEDM)[[1]](#footnote-1) of The Multilateral Interoperability Programme[[2]](#footnote-2) (MIP), is a good candidate for a common ground as being one of the most recognized and influential data models in the surveillance and situational awareness domains. The MIP organisation was established with a requirement to share relevant Command and Control information in a multinational or coalition environment. Recently the MIP has been working toward a new interoperability solution, called the MIP Information Model (MIM). The MIM approach enforces modularity and extensibility such that communities of interest can more easily exchange interoperability data sets that correspond to capability packages. Therefore, it is envisionable that a set of surveillance capability packages could be defined for maritime surveillance. The OASIS Emergency Data Exchange Language (EDXL)[[3]](#footnote-3) is another candidate standard, which is a suite of XML-based messaging standards that facilitate emergency information sharing between government entities and the full range of emergency-related organizations. Another relevant study is the PERSEUS Project (see Table in Section ), which builds a data model presenting comprehensive view of different types of information in order to create an integrated maritime surveillance space to support joint operations across borders. For sensor level, an approach from the OGC (SWE) provides a possible but not complete solution for sensor information models and services. OGC-SWE provides four mark-up languages for describing sensors, their capabilities and measurements, and accessing data and metadata for sensors. However, sustained interoperability based on XML and standardised tags in OGC-SWE are limited. Finally, the National Information Exchange Model (NIEM), originally developed by the United States Department of Homeland Security and the United States Department of Justice, now has evolved into an XML-based, data-centric information sharing standard for a large number of communities of interest including: immigration, intelligence, maritime, biometrics, international trade, justice and emergency management[[4]](#footnote-4). In early 2012, the NIEM took on an international dimension as Canadian government representatives started to promote the use of NIEM by Canada for the Law Enforcement and Public Safety sectors.

Having standards for both surveillance systems and sensor systems does not alone guarantee plug & play surveillance due to the fact that there are lots of standards, each having many versions and lacking common semantics. The information models proposed by standards are modular and support individual Community of Interest (COI) based data sets but the value of exchanged data depends on the availability of agreed formats – this is well provided by standards- and semantics so that it can be accessed from any source, interpreted, correlated, and subsequently integrated as part of subsequent workflows. For example, SensorML, which is a mark-up language to describe sensor devices and procedures to generate observations, is too ﬂexible and this allows sensors to be described in multiple ways and thus creates interoperability problems at the encoding level. As current standards generally are syntactic models, they do not provide facilities for abstraction, categorization, and semantic interoperability. Moreover, they do not allow reasoning on sensor data for surveillance applications. It is possible to create a semantically enriched version of the OGC-SWE, JC3IEDM or any other standard data model by automatic or semi-automatic transformation. For example, there are studies which create an OWL-DL representation of the JC3IEDM to enable automated reasoning for increased situational awareness[[5]](#footnote-5),[[6]](#footnote-6). However, these studies have two main weaknesses for current use. First, they refer to an old version of JC3IEDM. Second, they resulted with big ontologies, which confound many of the reasoners and exceeds their memory limitation.

Another relevant concept, on top of standards, is profiles. The profile concept covers choreographies, business rules and constraints. Choreographies can also be regarded as a step-by-step guideline to the surveillance processes. A machine computable process definition can be specified using one of the Business Process Specification and Choreography languages such as WSCDL (Web Service Choreography Description Language[[7]](#footnote-7)) or ebBP (ebXML Business Process[[8]](#footnote-8)). From the military perspective NATO provides some Standardization Agreements[[9]](#footnote-9) or studies such as NATO C3 Classification Taxonomy[[10]](#footnote-10) to define processes for surveillance functionalities to enable syntactic interoperability during collaborations; however these lacks semantics and they are not machine computable.

## Sensor Data Processing

The RECONSURVE project targeted fusion of the radar data with buoy-based directional acoustic data and UAV-based optical sensor data. The conducted experiments have shown promising results validating those data modalities as a viable source of information for reliable vessel detection. For this purpose a civilian ship image dataset construction effort has been carried out, and infrared images of various ships have been collected as part of the ITEA2 project RECONSURVE.

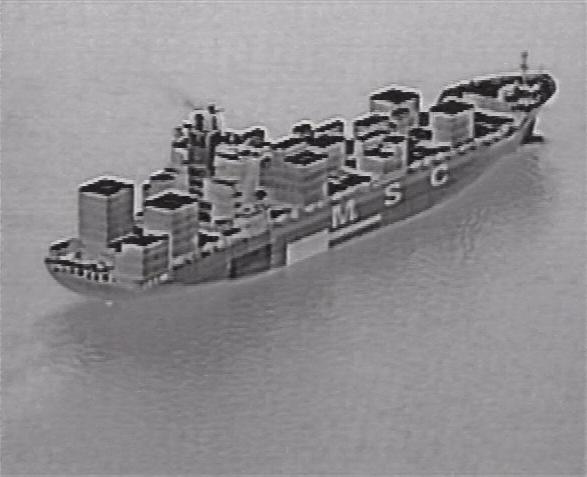


Figure 3 Images from RECONSURVE’s civilian ship image dataset

Vessel detection and classification are high level sensor processing operations which perform better when different low level sensor processing operations are fused. For instance, object tracking, image registration, camera motion detection algorithms serve the mentioned high level algorithms for better performance. In addition to fusing the signal in the sensor level as mentioned above, fusion of different low level algorithms is also aimed. Recently there has been an increasing interest on decision fusion systems, which combine the output of low and high level algorithms, in order to obtain better system solutions. This idea can also be extended to provide information for situational awareness as explained in the next section.

Another interesting piece of work has been done within the AMASS project, where the comprehensiveness and reliability of small vessel detection over the coastal line was claimed to be achieved by deployment of a network of unmanned buoys with optical and acoustic sensors mounted. It is also worth to mention the SEABILLA project, where radar data where complemented with a visual data from a network of unmanned aerial vehicles. At this moment, the usage of multi-modal sensors to achieve reliable and comprehensive maritime surveillance is highly fragmented due algorithmic (data fusion), architectural (interoperability) and institutional (multiple end-users) barriers. There are projects that aim at building up the mosaic from the individual technologies by focusing on both aspects of multi-modal sensing/fusion and plug-and-play interoperability.

Behaviour analysis and abnormality detection is currently investigated only at the research level, and is proposed to be solved by behavioural patterns methods. This is a high level task which requires a number of pre-processing steps (behaviour classification, pattern modelling and learning) and powerful AI algorithms to assess suspicious vessel-behaviour deviations in real-time. Due to the complexity of the methods, behavioural patterns are not yet deployed in the state-of-the-art maritime surveillance. The project enabled advanced content- and context-based analysis applications in surveillance and security.

## Behavioral Analysis

At present, the coastal and harbour surveillance systems use stationary radar data and Automatic Identification System (AIS) data obtained from vessels as main sources of information for monitoring. All vessels longer than 45m of length are supposed to cooperatively report their identification, location and route data through the AIS system, while smaller fishing vessels report via Vessel Monitoring System (VMS). The AIS and radar data are fused together for more comprehensive vessel tracking, navigation and collision avoidance. An example of such an approach to vessel behaviour analysis has been developed in the Dutch project Poseidon. Anomalies are detected here based on fused information from heterogeneous sources, a model of normal behaviour distilled from large amounts of trajectory data, and ontology of anomalous events.

Current maritime surveillance systems rely on cooperative behaviour of vessels. Especially this holds for small-size vessels, since coastal or harbour radars are not able to reliably detect small vessels in heavy-traffic areas or under rainy/windy weather conditions. In case a small vessel is not reporting itself and is moving in a close proximity with a large vessel, conventional surveillance systems are not able to detect and inspect the ship. This limitation is misused for illegal trafficking of drugs, humans and weapons in small boats. Moreover, this results in sudden collisions with larger vessels in crowded harbour areas due to the absence of external guidance from the coastal state.

In order to increase reliability and accuracy in vessel detection, several research projects/units proposed to complement the AIS/LRIT and radar data with sensors data of different modalities. The WATERVisie project (NL) proposed a solution for harbour surveillance based on fusion of the AIS/radar data with a data obtained from high-resolution optical sensors. The solution substantially increased reliability in detection of small non-cooperative vessels during day-light time and under good weather conditions. However, the solution is limited since at night-time and in foggy weather conditions the optical sensors are not able to provide sufficient data for reliable detection. The solution presupposes a line of sight. Typical AIS applications are as follows:

**Collision avoidance**

AIS is used in navigation primarily for collision avoidance. Due to the limitations of VHF radio communications, and because not all vessels are equipped with AIS, the system is meant to be used primarily as a means of lookout and to determine risk of collision rather than as an automated collision avoidance system, in accordance with the International Regulations for Preventing Collisions at Sea (COLREGS).

When a ship is navigating at sea, the movement and identity of other ships in the vicinity is critical for navigators to make decisions to avoid collision with other ships and dangers (shoal or rocks). Visual observation (unaided, binoculars, night vision), audio exchanges (whistle, horns, VHF radio), and radar or Automatic Radar Plotting Aid (ARPA) are historically used for this purpose. However, a lack of positive identification of the targets on the displays, and time delays and other limitation of radar for observing and calculating the action and response of ships around, especially on busy waters, sometimes prevent possible action in time to avoid collision.

While requirements of AIS are only to display a very basic text information, the data obtained can be integrated with a graphical electronic chart or a radar display, providing consolidated navigational information on a single display.

**Vessel traffic services**

In busy waters and harbors, a local Vessel Traffic Service (VTS) may exist to manage ship traffic. Here, AIS provides additional traffic awareness and provides the service with information on the type of other ships and their movement.

**Aids to navigation**

AIS was developed with the ability to broadcast positions and names of objects other than vessels, like navigational aid and marker positions. These aids can be located on shore, such as in a lighthouse, or on the water or on platforms. The US Coast Guard suggests that AIS might replace RACON, or radar beacons, currently used for electronic navigation aids.

The ability to broadcast navigational aid positions has also created the concepts of Synthetic AIS and Virtual AIS. In the first case, an AIS transmission describes the position of physical marker but the signal itself originates from a transmitter located elsewhere. For example, an on-shore base station might broadcast the position of ten floating channel markers, each of which is too small to contain a transmitter itself. In the second case, it can mean AIS transmissions that indicate a marker which does not exist physically, or a concern which is not visible (i.e., submerged rocks, or a wrecked ship). Although such virtual aids would only be visible to AIS equipped ships, the low cost of maintaining them could lead to their usage when physical markers are unavailable.

**Search and rescue**

For coordinating resources on scene of marine search and rescue operation, it is important to know the position and navigation status of ships in the vicinity of the ship or person in distress. Here AIS can provide additional information and awareness of the resources for on scene operation, even though AIS range is limited to VHF radio range. The AIS standard also envisioned the possible use on SAR Aircraft, and included a message (AIS Message 9) for aircraft to report position.

To aid SAR vessels and aircraft in locating people in distress a standard for an AIS-SART AIS Search and Rescue Transmitter is currently being developed by the International Electrotechnical Commission (IEC), the standard is scheduled to be finished by the end of 2008 and AIS-SARTs will be available on the market from 2009.

**Accident Investigation**

AIS information received by VTS is important for accident investigation to provide the accurate time, identity, position by GPS, compass heading, course over ground (COG), Speed (by log/SOG) and rate of turn (ROT) of the ships involved for accident analysis, rather than limited information (position, COG, SOG) of radar echo by radar. The maneuvering information of the events of the accident is important to understand the actual movement of the ship before accident, particularly for collision, grounding accidents. A more complete picture of the events could be obtained by Voyage Data Recorder (VDR) data if available and maintained on board for details of the movement of the ship, voice communication and radar pictures during the accidents. However, VDR data are not maintained due to the limited 12 hours storage by IMO requirement.

## State-of-the-Art in Communication and Networks

Multipath transport techniques like SCTP-CMT[[11]](#footnote-11) or Multipath-TCP[[12]](#footnote-12) enhance the capabilities of TCP and UDP by enabling concurrent data transport via multiple access technologies for multi-homed devices. More and more user devices support cellular and Wi-Fi access, possibly at the same time, this capability could be used to increase throughput as well as reduce load from congested cellular network, more and more mobile network operators offload user traffic in congested cells by shifting the user's traffic to Wi-Fi technology when possible[[13]](#footnote-13). The offloading is transparent to the user, which probably will not notice this, other than the device possibly using more battery since it is operating multiple radios at the same time.

Packet scheduling for multipath data transmissions can be achieved with basic mechanisms like round-robin, weighted round-robin, or fast adaptation weighted round robin. Accordingly, more information on the involved paths can be used to adapt the scheduling. Current scheduling algorithms for MPTCP distribute the data stream among the paths with respect to feedback about the queue length or round trip time (RTT). However, more detailed information on the path characteristics, such as bandwidth and loss estimations might enhance the performance of the multipath transmission. The current draft on SCTP concurrent multipath transmission still lacks a section covering appropriate multipath scheduling mechanisms[[14]](#footnote-14).

A major problem with multipath transmission is disordering. Disordering in networks can be investigated using typical queuing models like a GI/GI/n/m system with FIFO scheduling. Hereby, GI (General Independent) are arbitrary arrival and service processes, n is the number of service entities and m the queue length. For n > 1 packet reordering may occur due to different processing times of the packets. Different realizations of these abstract queuing models have been investigated in the literature[[15]](#footnote-15),[[16]](#footnote-16),[[17]](#footnote-17). These models allow an investigation of reordering issues for networks, based on statistical arrival and service processes for incoming packets. However, first steps for a computation of packet sequencing based on network monitoring have been presented in [[18]](#footnote-18),[[19]](#footnote-19).

In the case packet loss in one or multiple paths is not negligible; using one of the known TCP variants may not be adequate. For example, if the RTT is not short, waiting for a retransmission may not be possible in streaming video. Using FEC can be regarded as a solution but if not tuned well may result in large unnecessary overhead due to the redundant transmission of the FEC blocks. Thus, there is a need to design a new transport protocol that will work well in such an environment and adapt in the spirit of TCP, to the changing path conditions.

## Intelligent Interfaces

The increasing availability and maturity of the so-called Intelligent Agent technologies has led to great progress in the design and implementation of intelligent systems. In particular, agent technologies have been applied to human-machine interfaces (HMI) in order to create Intelligent User Interfaces (IUI). IUI allow humans to interact more easily with the system and therefore to focus on the tasks at hand. The visual component is the main component of the HMI, but can be complemented with tactile, voice, (keyboard) text and other human initiated inputs. Smartphones and tablet computers already provide rudimentary functions consistent with an enriched, facilitated human-machine interaction. However, the future Smart Surveillance Systems will require an increased level of functionality and also of coordination among the various complementary forms of human input.

Previous research has been conducted for over a decade in the area of intelligent information management aids for addressing the Cognitive and Machine Information Overload. For example, the concepts and ideas outlined in the Valued Information at the Right Time (VIRT) initiative can be utilized to implement adaptable notification and information-sharing strategies based on machine-computable triggers and events and thereby reduce the workload of system operators. The use of structured information repositories as part of the semantic interoperability layer, described below, is consistent with the use of the mechanisms defined as part of VIRT. The triggers and events can be configured based on user, organization, and context and thus can be adapted to different organizations and situations. Using VIRT can be applied to the controlled and automatic generation of warnings, alerts, and error messages and also to manage the information provided in dedicated situational awareness views. VIRT mechanisms can help to manage how information is provided to a specific user and also how information is shared across organizations. For example, an operator may want to know where all of the security officers assigned to a specific zone are located at all times. Conversely, this may only be required when an incident has occurred in the zone. Similarly, in the case of a significant threat detection, or a deflagration or hazardous material spill, it may be useful and appropriate to share information as quickly as possible across agencies. Today, there are no technological obstacles that prohibit the development of an automation interface to provide VIRT information sharing functionality for Smart Surveillance systems, as described in this proposal.

In the consumer electronics industry, Smartphones already are equipped with basic speech recognition technologies that provide some initial useful, although somewhat limited functionality. Clearly this is not sufficient for the purposes of Smart Surveillance system HMI. However, recent research in the area of Natural Language Processing (NLP) has set the stage for more advanced uses of NLP as part of intelligent interfaces that will allow the user to formulate voice-based queries and requests. For example, Watson[[20]](#footnote-20), developed by IBM, is an example of advanced NLP technologies that actually allow human users to query extensive information repositories and retrieve comprehensive responses that may include more than just a one-line answer. Watson is capable, today, of providing an answer to a query with references to the supporting evidence used to calculate an answer that also may include proposed actions.

The use of such powerful NLP technologies as part of Smart Surveillance systems there has a clear application: providing human operators a quick and efficient access to surveillance, security and other information. As a first step, NLP based technologies will be used essentially for the purposes of retrieving information in a timely and efficient and user-friendly manner. However, in the future, NLP also can be utilized as a decision-making aid. Research in the area of autonomous systems has indicated that when coupled with the use of intelligent agent technologies, Automation Management Strategies may be used to define how the decision-making responsibility is shared between humans and machines[[21]](#footnote-21). At the same time, parallel research in the areas of Computational Linguistics and Knowledge Representation already are providing the means to define formal languages (defined for the purpose of NLP) in order to connect human users to machine-based, structured information repositories. These languages allow for inference and reasoning; the machine deduction of new information based on existing information and make use of ontologies. Significant research already has been conducted in defining formal languages for NLP that can link the language syntax and structured, as derived from human speech acts, with the semantics of machine-readable dictionaries[[22]](#footnote-22).

Beyond very simple and limited translation mechanisms currently available to the public (e.g. Google Translation), current NLP technologies offer concrete solutions for multilingual functionalities. Once a machine-readable form has been calculated, translating an expression into other natural languages (for human consumption) is reduced to a simple task and thus provides an immediate benefit to users. In fact, advanced support for multiple languages already has been addressed in the multilingual extension to NLP known as MNLP. NLP frameworks with multilingual support already exist today and can be utilized to build intelligent interface components. This could contribute to eliminating language barriers and thus facilitate exchanges and collaboration across communities or nations that do not share a common language. Indeed the development of a robust voice-based information repository interface with multilingual capabilities represents an innovative exploitation of several related yet distinct technologies that are all reaching maturity at the same time. In parallel to the voice-based query, a traditional keyboard based query also should prove useful without introducing any additional obstacles.

# State of the Art in Standards

## OGC Sensor Web Enablement

This section describes the architecture implemented by Open Geospatial Consortium’s (OGC) [[23]](#footnote-23)Sensor Web Enablement Initiative (SWE)[[24]](#footnote-24). In much the same way that HTML and HTTP standards enabled the exchange of any type of information on the Web, the SWE initiative is focused on developing standards to enable the discovery of sensors and corresponding observations, exchange, and processing of sensor observations, as well as the tasking of sensors and sensor systems. The functionality that OGC has targeted within the Sensor Web includes:

* Discovery of sensor systems, observations, and observation processes that meet our immediate needs
* Determination of a sensor’s capabilities and quality of measurements
* Access to sensor parameters that automatically allow software to process and geo-locate observations
* Retrieval of real-time or time-series observations and coverages in standard encodings
* Tasking of sensors to acquire observations of interest
* Subscription to and publishing of alerts to be issued by sensors or sensor services based upon certain criteria

The Sensor Web represents a meta-platform that integrates arbitrary sensors and sensor networks; each maintained and operated by individual institutions, e.g. the Australian Water Resources Network, the European Environment Information and Observation Network, or the South African Earth Observation Network. This reflects the existing legal, organizational and technical situation. Sensors and sensor systems are operated by various organizations with varying access constraints, security, and data quality and performance requirements. The architectural design of the Sensor Web allows the integration of individual sensors as much as the integration of complete sensor systems without the need of fundamental changes to the legacy systems.

Within the SWE initiative, the enablement of such sensor webs is being pursued through the establishment of the following encodings for describing sensors and sensor observations,

1. Sensor Model Language (SensorML) – standard models and XML Schema for de-scribing sensors systems and processes; provides information needed for discovery of sensors, location of sensor observations, processing of low-level sensor observations, and listing taskable properties
2. Observations and Measurements Schema (O&M) – standard models and XML Schema for encoding observations and measurements from a sensor, both archived and real-time and through four standard interface definitions for Web services:
3. Sensor Observations Service (SOS) – standard web service interface for requesting, filtering, and retrieving observations and sensor system information
4. Sensor Planning Service (SPS) – standard web service interface for requesting user-driven acquisitions and observations, to (re-)calibrate a sensor or to task a sensor network
5. Sensor Alert Service (SAS) – standard web service interface for publishing and subscribing to alerts from sensors
6. Web Notification Services (WNS) – standard web service interface for asynchronous delivery of messages or alerts from any other web service

## Description Logics Reasoners

Currently, there are the following Description Logics reasoners in the literature: Racer Pro[[25]](#footnote-25), KAON2[[26]](#footnote-26), Fact++[[27]](#footnote-27) and Pellet[[28]](#footnote-28). A recent survey[[29]](#footnote-29) investigates the resoners considering their OWL support, correctness, efficiency, interface capabilities and inference services. The survey concludes that no system, except RacerPro and KAON2, is able to correctly solve at least those tests which lay within the language fragment that the tools claim to support in full. And to some extend KAON2 is not application ready since it fails very often with “out of memory errors” or requires significant processing time for language constructs, which are typically in real-world models such as cardinality restrictions. Pellet and FaCT++ do have some serious bugs which result in incorrect answers. In addition to the survey, in the scope of the thesis, the above mentioned reasoners are investigated in terms of their efficiency. Only Racer Pro could answer to the situational awareness ontology without “out of memory error”. Therefore, in this project Racer Pro is used as the Description Logics Reasoner and this reasoner will be used to develop/maintain the Situational Awareness Ontology.

## Joint Consultation, Command and Control Information Exchange Data Model (JC3IEDM)

Unilateral capability is important to nations but most planning is made on the assumption of alliance and coalition operations in scenarios that are difficult to predict and which often arise at short notice. Thus the nature and composition of a force structure to meet military requirements will be specific to requirement and based upon a general and flexible military capability. To achieve this, an assured capability for interoperability of information is essential. The successful execution of fast moving operations needs an accelerated decision-action cycle, increased tempo of operations, and the ability to conduct operations within combined joint formations. Commanders require timely and accurate information. Also, supporting command and control (C2) systems need to pass information within and across national and language boundaries. Moreover, tactical C2 information must be provided to the operational and strategic levels of command including other governmental departments. Additionally, forces must interact with non-governmental organizations, including international aid organizations. In this respect, the Multilateral Interoperability Programme (MIP)[[30]](#footnote-30) aims to deliver an assured capability for interoperability of information to support joint / combined operations.

The aim of the Multilateral Interoperability Programme (MIP) is to achieve international interoperability of Command and Control Information Systems (C2IS) at all levels from corps to battalion, or lowest appropriate level, in order to support multinational (including NATO), combined and joint operations and the advancement of digitization in the international arena.

Towards this aim, MIP produced Joint C3 Information Exchange Data Model (JC3IEDM)[[31]](#footnote-31) which is a model that when physically implemented aims to enable the interoperability of systems and projects required to share Command and Control (C2) information. JC3IEDM is an evolution of the C2IEDM[[32]](#footnote-32) standard that includes joint operational concepts, just as the Land Command and Control Information Exchange Data Model (LC2IEDM) was extended to become C2IEDM.

## Semantic Web Technologies

This part of the report provides a basic comprehension of Semantic Web and its latest technologies. Specifically, an examination of Web Ontology Language (OWL) and an evaluation of ontology tools are outlined in this section.

### Semantic Web

At a surprisingly accelerated rate, the Internet has become the central information station for individual to consume. The current web architecture contains vast amount of information resources that can be utilize in numerous beneficial manners. This web information is displayed with the aid of markup languages, such as HTML. Unfortunately, it is rendered solely for humans to translate and share data, rather than computer applications. It neither provides any meaning to this data nor help in understanding the content for computers to process. This data found on the Internet lacks structure and explicit meaning, creating difficulties for information retrieval and readability by computers. As a result, the true potential and effectiveness of the Internet is limited since it relies on human interpretation to use this information.

Therefore, the W3C organization has decided to pursue a “Semantic Web” project, where the Internet is transformed into a machine-interpretable network. It would contain “information [which has] well-defined meaning, better enabling computers and people to work in cooperation”. In this ‘semantic’ vision, the Internet would be extended with conceptual metadata[[33]](#footnote-33) that reveals the intended meaning of Web resources, making them more useful to machines. In other words, the documents are ‘marked up’ with semantic information so that the content of the document is easily interpreted by the computers. In a semantically enabled system, the tags in the document refer to defined concepts, and the system can parse the definition of the concept and use that to combine the information with other potentially related data in other applications. Subsequently, the software agents can apply this information to perform advanced tasks that humans may not be able to perform. It will “weave together an incredibly large network of human knowledge and will complement it with machine processability”. This process may ultimately create truly knowledgeable systems with various specialized reasoning services.

The Semantic Web addresses the shortcomings of the current web by offering a data centric markup language, XML, and the descriptive standards, RDF and OWL. eXtensible Markup Language (XML) provides a surface syntax for structured documents, but it does not provide sufficient data meaning for “efficient sharing of conceptualization”. In other words, it stores and displays information, however, it does not provide any description to it. Resource Description Framework (RDF) is a basic ontology language with graphical applications that combines XML syntax and semantics to represent information about resources on the web. Resources are described in terms of properties and property values using RDF statements. Due to RDF’s drawbacks listed below, this graphical language may not be a proper instrument for the Situational Awareness Ontology:

* It is weak to describe the resources in sufficient details, such as no localized range and domain constraints.
* It is difficult to provide reasoning support.

OWL (Web Ontology Language) has “more facilities, [such as additional vocabulary], for expressing meaning and semantics than XML, RDF, and RDF Schemas, and thus OWL goes beyond these languages in its ability to represent machine interpretable content on the Web”. In other words, OWL is a stronger language with greater machine interpretability and larger vocabulary than RDF. To facilitate the exchange of data between computer applications, standard vocabularies of a domain must be established and captured in an ontology. It is a knowledge representation model defined in terms of classes, properties and relationships for individuals who need to share information in a domain. Ontologies compose the primary foundation of OWL, hence it is vital to comprehend ontologies and its OWL components.

### Ontology

In computer science and information science, an ontology formally represents knowledge as a set of concepts within a domain, and the relationships between those concepts. It can be used to reason about the entities within that domain, and may be used to describe the domain. In theory, an ontology is a "formal, explicit specification of a shared conceptualization". An ontology renders shared vocabulary and taxonomy, which models a domain — that is, the definition of objects and/or concepts, and their properties and relations.

Ontologies are the structural frameworks for organizing information and are used in artificial intelligence, the Semantic Web, systems engineering, software engineering, biomedical informatics, library science, enterprise bookmarking, and information architecture as a form of knowledge representation about the world or some part of it. The creation of domain ontologies is also fundamental to the definition and use of an interoperability framework.

Contemporary ontologies share many structural similarities, regardless of the language in which they are expressed. As mentioned above, most ontologies describe individuals (instances), classes (concepts), attributes, and relations. In this section each of these components is discussed in turn. Common components of ontologies include:

* Individuals: instances or objects (the basic or "ground level" objects)
* Classes: sets, collections, concepts, classes in programming, types of objects, or kinds of things
* Attributes: aspects, properties, features, characteristics, or parameters that objects (and classes) can have
* Relations: ways in which classes and individuals can be related to one another
* Function terms: complex structures formed from certain relations that can be used in place of an individual term in a statement
* Restrictions: formally stated descriptions of what must be true in order for some assertion to be accepted as input
* Rules: statements in the form of an if-then (antecedent-consequent) sentence that describe the logical inferences that can be drawn from an assertion in a particular form
* Axioms: assertions (including rules) in a logical form that together comprise the overall theory that the ontology describes in its domain of application. This definition differs from that of "axioms" in generative grammar and formal logic. In those disciplines, axioms include only statements asserted as a priori knowledge. As used here, "axioms" also include the theory derived from axiomatic statements
* Events: the changing of attributes or relations

Ontologies are commonly encoded using ontology languages such as OWL and RDFS. OWL is W3C’s latest Semantic technology that builds these ontologies to enable agents to exchange data across web applications and resources.

### Web Ontology Language (OWL)

Released in February 2004 by the W3C, Web Ontology Language (OWL) is an ontology language that describes the classes, properties and relations between them that are inherent in Web documents and resources. Jim Hendler and Guus Schreiber, co-chairs for the Web Ontology Working Group, describe OWL as the following:

*"OWL takes a major step forward in representing and organizing knowledge on the World Wide Web. It strikes a sound balance between the needs of industry participants for a language which addresses their current Web use cases, and the restrictions on developing an ontology language that meshed with established scientific principles and research experience."*

OWL is used to describe, share and publish the set of terms that are inherent in Web documents and applications. OWL uses both Unique Resource Locators (URL)[[34]](#footnote-34) (e.g.: <http://www.w3c.org>) for naming and the description framework for the Web provided by RDF to extend the capabilities of ontologies. OWL is a vocabulary extension of RDF and RDF-S by providing an elaborated description of classes, properties, and individuals. This feature enhances the machine interpretability of Web content. OWL is derived from two other languages, DAML (DARPA Agent Markup Language) & OIL (Ontology Inference Layer).

OWL has three sub languages, each with a different level of expressive description of the data:

* OWL Lite: It is the simplest language for ontologies with simple class hierarchies and constraints. This subset of OWL-DL contains an easier reasoner than the other species.
* OWL-DL: It corresponds to Description Logics[[35]](#footnote-35), meaning that it has “decidable reasoning”. Thus, it automatically computes the classification hierarchy and checks for inconsistencies. OWL-DL does not allow datatype properties to be transitive, symmetric, or have inverse properties. Therefore, relationships can only be formed between individuals or between an individual and a data value.
* OWL Full: It is an extension of RDF with OWL syntax, where it allows for classes as instances. In OWL-Full, classes can be related, but this cannot be reasoned with.

## Web Services

A Web service is in fact a RPC (remote procedure call); however the vital point is that the entire world has agreed on the standard. The standard is based on the SOAP and XML-messaging. Thus, the web services technologies do not depend on any operating systems or platform. SOAP forms the layer for the messaging and XML is the formats of the messages. The WSDL file contains the technical description of the web services, such as input and output formats.

Interactions among Web services involve three types of participants: service provider, service registry and service consumer. Service registries are searchable repositories of Web Service descriptions. There are two well-known service registries: Electronic Business XML (ebXML[[36]](#footnote-36)) Registries and the Universal Description, Discovery, Integration framework (UDDI[[37]](#footnote-37)) Registries.

In addition to building web services interfaces to existing applications, there must also be a standard approach to connecting these web services together to form more meaningful business processes. The ability to integrate and assemble individual Web services into standards-based business processes is an important element of the service-oriented enterprise and the overall Web service technology “stack.”

The term Web services has gained a lot of momentum. Many software vendors (large and small) are announcing Web services initiatives and adoption. Many organizations are involved in the refinement of Web services standards. Although there seems to be a slow convergence towards a common understanding of what the term means, there is no single, universally adopted definition of what is meant by the term Web service. This situation is reminiscent of the early days of object-oriented programming: Not until the concepts of inheritance, encapsulation, and polymorphism were well defined did object-oriented programming become accepted into the mainstream of development methodologies.

Several major Web services infrastructure providers have published their definitions for a Web service:

IBM offers this definition at:

“A Web service is an interface that describes a collection of operations that are network accessible through standardized XML messaging. Web services fulfill a specific task or a set of tasks. A Web service is described using a standard, formal XML notion, called its service description that provides all of the details necessary to interact with the service, including message formats (that detail the operations), transport protocols, and location.”

The nature of the interface hides the implementation details of the service so that it can be used independently of the hardware or software platform on which it is implemented and independently of the programming language in which it is written. This allows and encourages Web services based applications to be loosely coupled, component-oriented, cross-technology implementations. Web services can be used alone or in conjunction with other Web services to carry out a complex aggregation or a business transaction.

Microsoft has a couple of definitions for Web service:

“A Web service is a unit of application logic providing data and services to other applications. Applications access Web services via ubiquitous Web protocols and data formats such as HTTP, XML, and SOAP, with no need to worry about how each Web service is implemented. Web services combine the best aspects of component-based development and the Web, and are a cornerstone of the Microsoft .NET programming model.”

A Web service is programmable application logic accessible using standard Internet protocols. Web services combine the best aspects of component-based development and the Web. Like components, Web services represent black-box functionality that can be reused without worrying about how the service is implemented. Unlike current component technologies, Web services are not accessed via object-model-specific protocols, such as the distributed Component Object Model (DCOM), Remote Method Invocation (RMI), or Internet Inter-ORB Protocol (IIOP). Instead, Web services are accessed via ubiquitous Web protocols and data formats, such as Hypertext Transfer Protocol (HTTP) and Extensible Markup Language (XML). Furthermore, a Web service interface is defined strictly in terms of the messages the Web service accepts and generates. Consumers of the Web service can be implemented on any platform in any programming language, as long as they can create and consume the messages defined for the Web service interface.

Sun provides the following definition at:

“Web services are software components that can be spontaneously discovered, combined, and recombined to provide a solution to the user's problem/request. The Java™ language and XML are the prominent technologies for Web services.”

It should be noted that there is broad agreement on what a Web service might be, but no single agreed-upon definition. Many developers will claim that they cannot define what a Web service is, but they know one when they see one.

In general, a Web service is a platform and implementation independent software component that can be:

* + Described using a service description language
  + Published to a registry of services
  + Discovered through a standard mechanism (at runtime or design time)
  + Invoked through a declared API, usually over a network
  + Composed with other services

One important point is that a Web service need not necessarily exist on the World Wide Web. This is an unfortunate historical naming issue. A Web service can live anywhere on the network, Inter- or intranet; some Web services can be invoked by a simple method invocation in the same operating system process, or perhaps using shared memory between tightly coupled processes running on the same machine. In fact, Web services have little to do with the browser-centric, HTML-focused World Wide Web. Sometimes, the names we choose in the information technology (IT) industry don't make a lot of sense; they simply take on a life of their own.

Another important point is that a Web service's implementation and deployment platform details are not relevant to a program that is invoking the service. A Web service is available through its declared API and invocation mechanism (network protocol, data encoding schemes, and so on). This is analogous to the relationship between a Web browser and a Web application server: Very little shared understanding exists between the two components. The Web browser doesn't particularly care if the Web application server is Apache Tomcat, Microsoft IIS, or IBM Websphere. The shared understanding is that they both speak HTTP and converse in HTML or a very limited set of MIME types. Similarly, the Web application server really doesn't care what kind of client is using it—various brands of Web browsers or even non-browser clients. This minimal shared understanding between components allows Web services to form a system of loosely coupled components.

To a business person, the Web services approach is all about integration: integrating application functionality within an organization or integrating applications between business partners (in a supply chain, for example). The important point is that application integration is enabled without tight lock-in to any particular business partner. If another supplier has a better price, shipping terms, or quality assurance, then a company's reorder systems can be easily repositioned to use that supplier; doing so is as easy as pointing a Web browser at a different Web site. With a broader adoption of Web services and XML document format standards, this style of dynamic business partner integration will become more broadly used.

When systems are this easy to integrate, an organization's reach to suppliers, customers, and other business partners is extended, yielding cost savings, flexible business models, better customer service, higher customer retention, and so on. Just as IT is fundamental to the efficient operations of an organization, Web services-based systems integration will be fundamental to flexible, lightweight systems integration—for internal application integration within an organization over an intranet and external partner integration over the Intranet or extended virtual private network. So, from a business perspective, a Web service is a business process or step within a business process that is made available over a network to internal and/or external business partners to achieve some business goal.

From a technical perspective, a Web service is nothing more than a collection of one or more related operations that are accessible over a network and are described by a service description. At this level, the Web services concept is not new. With Web services, the IT industry is trying to address the fundamental challenge of distributed computing that has been around for decades locating and accessing remote systems. The big difference is that now the industry is approaching this problem using open technology (XML and Internet protocols) and open standards managed by broad consortia such as the World Wide Web Consortium (W3C, which manages the evolution of the SOAP and WSDL specifications).

## Data Distribution Services (DDS)

Many distributed applications exist today and many more are being planned for the future. One requirement common to all distributed applications is the need to pass data between different threads of execution. These threads may be on the same processor, or they may spread across different nodes. We may also have a combination of these: multiple nodes with multiple threads or processes working on each one. Each of these nodes or processes is connected through a transport mechanism such as Ethernet or shared memory. Basic protocols such as TCP/IP or higher level protocols such as HTTP can be used to provide standardized communication paths between each of the nodes. One mechanism that can be used to facilitate this data communication path is the Data Distribution Service for Real Time Systems, known as DDS.

Data Distribution Service is introduced in 2004 by Object Management Group, the company that governs specifications for CORBA, UML and many other standards. DDS is the first open international middle-ware standard directly addressing *publish-subscribe* communications for *real-time and embedded systems*. It standardizes the software application programming interface by which a distributed application can use “Data-Centric Publish-Subscribe”, the DCPS layer of DDS, as a communication mechanism. Since DDS is implemented as an infrastructure solution, it can be added as the communication interface for any software application. Today, DDS is recommended by key administration worldwide and it is widely adopted by several different application areas such as automated trading, simulations, telemetry etc. the reasons that make OMG’s Data Distribution Service as preferable as it is now can be listed as follows:

* Based on a simple Publish-Subscribe mechanism
* Flexible and adaptable architecture
* Low overhead, high-performance systems
* Deterministic data delivery
* Dynamically scalable
* Efficient use of transport bandwidth
* Usable for one-to-one, one-to-many, many-to-many communications
* Large number of quality of service (QoS) parameters that give publishers and subscribers complete control of the features of the data in the system

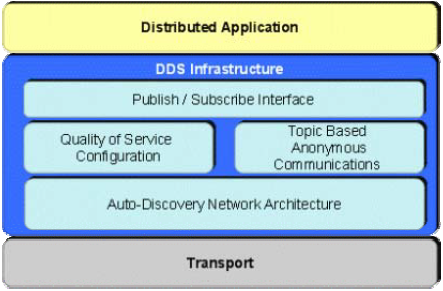


Figure 4 The structure of Data Distribution Service

## UN/CEFACT Unqualified Data Types

United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT[[38]](#footnote-38)) defined and published a catalog of data types to be used in common business document. They are generic in that they can be used as the data type of any information element. In RECONSURVE, in order to increase interoperability, these unqualified data types are adopted. The list of data types are as follows:

Table 1 Unqualified Data Types

| Primary *Representation Term* | Definition |
| --- | --- |
| Amount | A number of monetary units specified in a currency where the unit of currency is explicit or implied. |
| Binary Object | A set of finite-length sequences of binary octets.  [Note: This *Representation Term* shall also be used for *Data Types* representing graphics (i.e. diagram, graph, mathematical curves, or similar representation), pictures (i.e. visual representation of a person, object, or scene), sound, video, etc.] |
| Code | A character string (letters, figures or symbols) that for brevity and / or language independence may be used to represent or replace a definitive value or text of a *Property*.  [Note: The term 'Code' should not be used if the character string identifies an instance of an *Object Class* or an object in the real world, in which case the *Representation Term* identifier should be used.] |
| Date Time | A particular point in the progression of time (ISO 8601).  [Note: This *Representation Term* shall also be used for *Data Types* only representing a Date or a Time.] |
| Identifier | A character string used to establish the identity of, and distinguish uniquely, one instance of an object within an identification scheme from all other objects within the same scheme. |
| Indicator | A list of exactly two mutually exclusive Boolean values that express the only possible states of a *Property*.  [Note: Values typically indicate a condition such as on/off; true/false etc.] |
| Measure | A numeric value determined by measuring an object. Measures are specified with a unit of measure. The applicable unit of measure is taken from UN/ECE Rec. 20.  [Note: This *Representation Term* shall also be used for measured coefficients (e.g. m/s).] |
| Numeric | Numeric information that is assigned or is determined by calculation, counting or sequencing. It does not require a unit of quantity or a unit of measure.  [Note: This *Representation Term* shall also be used for *Data Types* representing Ratios (i.e. rates where the two units are not included or where they are the same), Percentages, etc.) |
| Quantity | A counted number of non-monetary units. Quantities need to be specified with a unit of quantity.  [Note: This *Representation Term* shall also be used for counted coefficients (e.g. flowers/m²).] |
| Text | A character string (i.e. a finite set of characters) generally in the form of words of a language.  [Note: This *Representation Term* shall also be used for names (i.e. word or phrase that constitutes the distinctive designation of a person, place, thing or concept).] |

## OASIS Common Alerting Protocol (CAP)

The Common Alerting Protocol (CAP)[[39]](#footnote-39) is an XML-based data format for exchanging public warnings and emergencies between alerting technologies. CAP allows a warning message to be consistently disseminated simultaneously over many warning systems to many applications. CAP increases warning effectiveness and simplifies the task of activating a warning for responsible officials. CAP also facilitates the detection of emerging patterns in local warnings of various kinds. And CAP provides a template for effective warning messages based on best practiced identified in academic research and real-world experience.

CAP provides an open, non-proprietary digital message format for all types of alerts and notifications. It does not address any particular application or telecommunications method. The CAP format is compatible with emerging techniques, such as Web services, existing formats including the Specific Area Message Encoding (SAME) user for the US National Oceanic and Atmospheric Administration (NOAA) Weather Radio and the Emergency Alert System(EAS) while offering enhanced capabilities that include:

* Flexible geographic targeting using latitude/longitude shapes and other geospatial representations in three dimensions;
  + Multilingual and multiaudience messaging;
  + Phased and delayed effective time and expiration;
  + Enhanced message update and cancellation features;
  + Template support for framing complete and effective warning messages;
  + Compatible with digital encryption and signature capability; and,
  + Facility for digital images and audio

Key benefits of CAP include reduction of costs and operational complexity by eliminating the need for multiple custom software interfaces to the many warning sources and dissemination systems involved in all-hazard warning. The CAP message format can be converted to and from all kinds of sensor and alerting technologies.

# Conclusion

This report presents the current state of the art in the standards and specifications that are related to the scope of RECONSURVE Project. The standards reviewed in this document is used in the following components of the RECONSURVE project as described below:

* OWL DL served as the format of the Situational Awareness Ontology.
* Racer Pro description logics reasoner was used to maintain the Situational Awareness Ontology.
* The situational awareness rules were defined in SWRL syntax and they were converted to either Jess or Drools for execution.
* JC3IEDM was the basis for the Situational Awareness Ontology and this standard was used in the interoperability of C2 Systems.
* Web Services was used to gather information from external organizations. This information was mainly be used to execute the situational awareness rules.
* DDS was the mechanism to communicate with C2 systems.
* OASIS CAP was used to send alarms to both C2 systems and external organizations.
* UN/CEFACT Unqualified Data Types constituted the data type space of the information items exchanged in the RECONSURVE system.
* OGC SWE specifications were used to retrieve information from sensors in a standard way.
* AIS was used to identify the location, name, direction of the vessels.

1. JC3IEDM: http://www.mip-site.org/publicsite/04-Baseline\_3.0/JC3IEDM-Joint\_C3\_Information\_Exchange\_Data\_Model/ [↑](#footnote-ref-1)
2. Multilateral Interoperability Programme (MIP): MIP Website. https://mipsite.lsec.dnd.ca/Pages/Default.aspx. [↑](#footnote-ref-2)
3. OASIS Emergency Management TC, http://www.oasis-open.org/committees/tc\_home.php?wg\_abbrev=emergency [↑](#footnote-ref-3)
4. For a complete list of current communities of interest, see: https://www.niem.gov/communities/Pages/communities.aspx [↑](#footnote-ref-4)
5. C. Matheus and B. Ulicny, On the Automatic Generation of an OWL Ontology based on the Joint C3 Information Exchange Data Model. 12th International Command and Control Research and Technology Symposium, Newport, RI, June 19-21, 2007. [↑](#footnote-ref-5)
6. S. Wartik, A JC3IEDM OWL-DL Ontology, OWLED 2009 OWL: Experiences and Directions, Sixth International Workshop Chantilly, Virginia, USA 23-24 October 2009 [↑](#footnote-ref-6)
7. <http://www.w3.org/TR/2004/WD-ws-cdl-10-20041217/> [↑](#footnote-ref-7)
8. <http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ebxml-bp> [↑](#footnote-ref-8)
9. http://nsa.nato.int/nsa/nsdd/listpromulg.html [↑](#footnote-ref-9)
10. AC/322-N(2012)0092-AS1, C3 Classification Taxonomy, 19 June 2012 [↑](#footnote-ref-10)
11. J. Iyengar, P. Amer, and R. Stewart. Concurrent Multipath Transfer Using SCTP Multihoming. SPECTS’04, 2004. [↑](#footnote-ref-11)
12. A. Ford, C. Raiciu, M. Handley, S. Barre, J. Iyengar. Architectural Guidelines for Multipath TCP Development. IETF RFC 6182, March 2011. [↑](#footnote-ref-12)
13. GSM Association. Wi-Fi Offload whitepaper. version 1.0., 19 April 2010. [↑](#footnote-ref-13)
14. M. Becke, T. Dreibholz, J. Iyengar, P. Natarajan, M. Tuexen. Load Sharing for the Stream Control Transmission Protocol (SCTP). July 2011, Expires January , 2012. [↑](#footnote-ref-14)
15. Ye Xia and David Tse. Analysis on Packet Resequencing for Reliable Network Protocols. Infocom 2003. [↑](#footnote-ref-15)
16. G. Harrus and B. Plateau. Queueing Analysis of a Reordering Issue. IEEE Trans. Softw. Eng. March 1982. [↑](#footnote-ref-16)
17. T. Yum and T. Ngai. Resequencing of Messages in Communication Networks. IEEE Tran. On Communication. February 1986. [↑](#footnote-ref-17)
18. T. Zinner, K.Tutschku, A. Nakao and P. Tran-Gia. Using Concurrent Multipath Transmission for Transport Virtualization: Analyzing Path Selection. ITC 2010. [↑](#footnote-ref-18)
19. T. Zinner, K.Tutschku, A. Nakao and P. Tran-Gia. Performance Evaluation of Packet Re-ordering on Concurrent Multipath Transmissions for Transport Virtualization. JCNDS Special Issue 2011. [↑](#footnote-ref-19)
20. http://www-03.ibm.com/innovation/us/watson/ [↑](#footnote-ref-20)
21. http://scholar.lib.vt.edu/ejournals/JOTS/v32/v32n1/cummings.html [↑](#footnote-ref-21)
22. http://lemon-model.net/ [↑](#footnote-ref-22)
23. <http://www.opengeospatial.org/> [↑](#footnote-ref-23)
24. <http://www.opengeospatial.org/projects/groups/sensorweb> [↑](#footnote-ref-24)
25. <http://www.racer-systems.com/> [↑](#footnote-ref-25)
26. <http://kaon2.semanticweb.org/> [↑](#footnote-ref-26)
27. <http://owl.man.ac.uk/factplusplus/> [↑](#footnote-ref-27)
28. <http://clarkparsia.com/pellet/> [↑](#footnote-ref-28)
29. Thorsten Liebig, “Reasoning with OWL - System Support and Insights”, Technical Report,

    September 2006. [↑](#footnote-ref-29)
30. <http://www.mip-site.org/> [↑](#footnote-ref-30)
31. <http://www.mip-site.org/publicsite/04-Baseline_3.0/JC3IEDM-Joint_C3_Information_Exchange_Data_Model/> [↑](#footnote-ref-31)
32. <http://www.mip-site.org/publicsite/03-Baseline_2.0/C2IEDM-C2_Information_Exchange_Data_Model/> [↑](#footnote-ref-32)
33. Metadata is “data about data”, meaning a set of facts about the content. [↑](#footnote-ref-33)
34. URI is an address for a resource available in the Web . [↑](#footnote-ref-34)
35. Description Logic focuses on descriptions to express logic (such as union, intersection and negation) of a domain. It emphasizes on the use of classification and subsumption reasoning for inference [11]. [↑](#footnote-ref-35)
36. www.ebxml.org [↑](#footnote-ref-36)
37. www.uddi.org [↑](#footnote-ref-37)
38. <http://www.unece.org/cefact/about.htm> [↑](#footnote-ref-38)
39. E.Jones & A.Botterell(Eds.).(2005).Common Alerting Protocol,v.1.1. Retrieved 27th July 2011 from <http://www.oasis-open.org/committees/download.php/15135/emergency-CAPv1.1-Corrected_DOM.pdf> [↑](#footnote-ref-39)