Information Process Architecture Volume 2: Survey of IPA Like Systems

Document WINNF-09-P-0021

Version V1.0.0

27 June 2012

TERMS, CONDITIONS & NOTICES

This document has been prepared by the Cognitive Radio Work Group to assist The Software Defined Radio Forum Inc. (or its successors or assigns, hereafter "the Forum"). It may be amended or withdrawn at a later time and it is not binding on any member of the Forum or of the Cognitive Radio Work Group.

Contributors to this document that have submitted copyrighted materials (the Submission) to the Forum for use in this document retain copyright ownership of their original work, while at the same time granting the Forum a non-exclusive, irrevocable, worldwide, perpetual, royalty-free license under the Submitter's copyrights in the Submission to reproduce, distribute, publish, display, perform, and create derivative works of the Submission based on that original work for the purpose of developing this document under the Forum's own copyright.

Permission is granted to the Forum's participants to copy any portion of this document for legitimate purposes of the Forum. Copying for monetary gain or for other non-Forum related purposes is prohibited.

THIS DOCUMENT IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY USE OF THIS SPECIFICATION SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE FORUM, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS DOCUMENT.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the specification set forth in this document, and to provide supporting documentation.

This document was developed following the Forum's policy on restricted or controlled information (Policy 009) to ensure that that the document can be shared openly with other member organizations around the world. Additional Information on this policy can be found here: http://www.wirelessinnovation.org/page/Policies_and_Procedures

Although this document contains no restricted or controlled information, the specific implementation of concepts contain herein may be controlled under the laws of the country of origin for that implementation. Readers are encouraged, therefore, to consult with a cognizant authority prior to any further development.

Wireless Innovation Forum ™ and SDR Forum ™ are trademarks of the Software Defined Radio Forum Inc.

Table of Contents

TERMS, CONDITIONS & NOTICES i
Executive Summary of Document v
Contributions vi
1 Motivation, Methodologies, and Background to Survey 1
  1.1 The Importance of Collaborating Intelligent Information Systems 1
  1.2 IPA Volume 1 Overview 1
    1.2.1 The Information System Framework 2
    1.2.2 The Information System Structure 2
    1.2.3 Information System Transaction Cycle and Modeling Components 3
  1.3 Issues Addressed by this Document 4
  1.4 Remaining Document Organization 4
  1.5 Introduction References 4
2 Cursor on Target 5
  2.1 System Objectives 6
  2.2 System Design 7
    2.2.1 Architecture 7
    2.2.2 CoT XML Schema 7
    2.2.3 CoT Conventions and Functions 11
  2.3 Lessons Learned from CoT for IPA 11
  2.4 Cursor on Target References 12
3 Standard Spectrum Resource Format 13
  3.1 System Objectives 16
  3.2 System Design 17
    3.2.1 Architecture 17
    3.2.2 Basic Message Format 18
  3.3 Common Element Inheritance 19
  3.4 Values 20
  3.5 Classification 23
  3.6 Time 23
  3.7 Handling Contextual Disagreements 23
  3.8 Radio Domain Information 24
  3.9 Lessons Learned from SSRF for IPA 24
  3.10 Standard Spectrum Resource Format References 25
4 1900.6 26
  4.1 System Objectives 26
  4.2 System Design 26
    4.2.1 Architecture 27
    4.2.2 Service Access Points 28
    4.2.3 1900.6 Services 30
    4.2.4 Information Description 32
  4.3 State Diagrams 37
  4.4 Lessons Learned from 1900.6 for IPA 39
  4.5 IEEE 1900.6 References 40

5 Extensible Markup Language (XML) ... 40
  5.1 XML Objectives ... 40
  5.2 XML Structure ... 41
  5.3 Lessons Learned from XML for IPA ... 43
  5.4 XML References ... 44
6 Modeling Language for Mobility (MLM) ... 44
  6.1 Goals ... 44
  6.2 CRO: Cognitive Radio Ontology ... 45
  6.3 A Case Study: MLM Based Link Optimization ... 48
  6.4 Lessons Learned ... 51
  6.5 MLM References ... 53
7 Survey Insights and Future Work ... 53
  7.1 Common Threads from Surveyed Protocols ... 54
  7.2 Recommendations ... 54
  7.3 Future Work ... 57
Appendix A: Acronym List ... 59

List of Figures

Figure 1: Information System Structure. From [IPA_10] ... 2
Figure 2: Information System Flow Model with Context. From [IPA_10] ... 3
Figure 3: FalconView: One of the more popular viewers [Konstantopoulos_06] ... 5
Figure 4: CoT XML Schema Base Schema [Kristan_09]. ... 8
Figure 5: Example type hierarchies [Kristan_09] ... 9
Figure 6: Example subschema. From [Kristan_09] ... 10
Figure 7: Example Cot message reporting a hostile fixed wing air craft at the latitude and longitude specified, but only valid at the time of reporting. From [Kristan_09]. ... 11
Figure 8: Warfighter Spectrum Use Below 40 GHz From: http://dodreports.com/pdf/ada476476.pdf. ... 14
Figure 9: From Figure 1.4.3 in [MCEB_09] ... 16
Figure 10: SSRF Manager Core Tasks. From Figure 2.1.1 in [MCEB_09] ... 17
Figure 11: Basic Information Exchange between two SSRF-based tools. From Figure 1.4.1 in [MCEB_09] ... 17
Figure 12: SSRF expects translation processes to be used to communicate with non-SSRF compliant systems while SSRF XML exchange is used elsewhere. From Figure 1.1.1 in [MCEB_09] ... 18
Figure 13: Top Level Elements in SSRF. From Figure II-2 in [MCEB_09] ... 19
Figure 14: Attributes of the Common Element in SSRF. From [MCEB_09] ... 20
Figure 15: Waveform Descriptive Information in SSRF for transmit emissions. From Figure II-12.h in [MCEB_09]. ... 24
Figure 16: 1900.6 System Model. From Figure 1 in [1900.6] ... 28
Figure 17: Example implementation of 1900.6 in station management entity in control plane. From Figure 9 in [1900.6] ... 29
Figure 18: Command Group Structure. From Figure 10 in [1900.6] ... 32
Figure 19: Sensing Control Class. From Figure 13 in [1900.6] ... 35

Figure 20: State diagram for sensing related information exchange. From Figure 17 in [1900.6] 38
Figure 21: Illustrating the basic elements of an XML element. Modified from an example in [XML_tutorial] 42
Figure 22: Top Level Portion of CRO. [Li_09] 47
Figure 23: A small part of the CRO ontology 48
Figure 24: Inference Engine, Ontology and Policy 50
Figure 25: Architecture of Ontology-Based Radio 50
Figure 26: Implementation Results 51
Figure 27: An overall communications context could be synthesized by piecing together schemas at different layers and functionalities. But this is easiest to maintain and extend when the schema are orthogonal. 56

List of Tables

Table 1: Information System Framework Components. From [IPA_10] 2
Table 2: Key 1900.6 Terminology. Excerpted from Section 3 in [1900.6] 27
Table 3: Use Cases for Interfaces in 1900.6. Excerpted from Section 1 in [1900.6] 28
Table 4: SAPs and Service Classes in 1900.6 30
Table 5: Units used in 1900.6. From Table 1 in [1900.6] 33
Table 6: Classes of Information defined in [1900.6] 34
Table 7: PerfMetric Parameters 36
Table 8: TimeSync Parameters 36
Table 9: DataSheet Parameters 37
Table 10: States defined in 1900.6 38

Executive Summary of Document

Information processing systems are an increasingly important part of modern communications and society. As these systems are enhanced with artificial intelligence techniques to enable cognitive radio, dynamic spectrum access, smart grid, and many other applications, it will be important that these new intelligent processes be capable of understanding the context and meaning of the information on which they operate. As part of the Information Process Architecture project's study into how information processes are remaking communications systems and society, impacting interoperability, and enhancing extensibility, this document surveys existing information exchange protocols to gather insights into how to effectively design information transfer mechanisms in communications systems to facilitate accurate and meaningful adaptations by intelligent agents embedded in those communications systems.

The document surveys the following information processing protocols:

- **1900.6** – an IEEE standard for exchanging sensor related information in support of DSA applications
- **Cursor on Target** – an Air Force / MITRE originated standard to facilitate the exchange of actionable situational awareness information between autonomous systems and humans
- **Meta-Language for Mobility** – a language developed within the Wireless Innovation Forum being considered as the basis for IEEE 1900.5 that allows cognitive radios to understand the meaning of wireless domain information
- **Standard Spectrum Resource Format** – a military standard for expressing the capabilities and requirements of wireless systems in a machine interpretable standard
- **XML** – an extensible standard for providing information in context in a machine and human readable format that provides the foundation for many modern application layer communications schemes

Each surveyed protocol is briefly overviewed and then reviewed for insights that can be applied to the goal of facilitating actionable, reliable information for intelligent processes embedded in wireless communications systems. The document concludes with a list of recommendations for developing an extensible information exchange framework for cognitive radio applications.

Contributions

This document was drafted by the Cognitive Radio Work Group of the Wireless Innovation Forum and received valuable contributions from the following:

Ihsan Akbar, *Harris*

Kuan Collins, *SAIC*

Daniel Devasirvatham, *SAIC*

Pete Cook, *Hypres*

Mitch Kokar, *Vistology*

Vince Kovarik, *PrismTech*

Neal Mellen, *TDK*

Joe Mitola, *Stevens Institute*

James Neel, *CRT*

Bob Schutz, *Datron*

John Stine, *MITRE*

Darcy Swain, *MITRE*

Bruce Oberlies, *Motorola Solutions*

Jason Ferguson, *Subject Matter Expert*

Information Process Architecture Volume 2: Survey of IPA-like Systems

1 Motivation, Methodologies, and Background to Survey

1.1 The Importance of Collaborating Intelligent Information Systems

As an increasingly larger number of "intelligent" information systems are being deployed, researchers are beginning to turn to ways in which different intelligent systems, perhaps designed with different technologies or different intended user groups and applications, can leverage the services and information contained in other information systems. By allowing one intelligent or information system or agent to leverage the capabilities of another intelligent information system, efficiencies can be gained as redundancies are eliminated and an intelligent system can bring to bear far more actionable information than it could reasonably gather or process on its own.

An initial example of this train of thought for wireless systems is the current solution that the US (and now many other) regulators converged upon to better utilize spectrum by allowing secondary devices to access the "white spaces" available within TV spectrum. [FCC_02] Initial tests of prototype radio attempting to discern the presence of protected signals in the TV spectrum (TV transmissions and wireless microphone signals) were a disappointment. [OET_08] These tests led to the conclusion that, at least for now devices would be unable to reliably gather sufficient information to provide an acceptable level of protection to the primary spectrum incumbents when acting on information that it was able to gather on its own. [FCC_08]

To enable secondary transmissions in the TV bands, an alternate information source was required: primarily Federal Communications Commission (FCC) geographical registrations of TV towers and other protected services. This information would be made available to TV Band Devices (TVBDs) via a database to enable selection of available TV channels in a specific geographic area to be used for secondary communication services. [FCC_08] This is a perfect example of the need for collaborating intelligent information systems.

1.2 IPA Volume 1 Overview

Information Process Architecture (IPA) is a project started in 2009 within the Wireless Innovation Forum (WInnF) to explore the architecture of converged Information and Telecommunication Systems (IT) that are required to better utilize scarce spectrum resources. The IPA is focusing on better understanding how information processing systems can be designed to facilitate the transfer of data between communications systems where systems are frequently developed and implemented independently and yet are fundamentally similar.

The first report produced by the IPA project included the following results which are described in more detail in the following three sections: [IPA_10]

- Information System Framework – a formalized decomposition of the forces that shape the design and operation of information systems
- Information System Structure – a generalized model that describes how application processes (both user controlled and autonomous) make use of data communications, data management and storage, and system services to implement an information system

- Information System Transaction Cycle and Modeling Components – conceptual model of how information (data with appropriate context) is conveyed from a sender to a recipient with stylized pictograms for rapid graphical representation of the operation of inter-connected information systems.

1.2.1 The Information System Framework

The Information System Framework decomposes the forces that shape the design and operation of an information system into six components as shown in Table 1.

Table 1: Information System Framework Components. From [IPA_10]

Framework Component	Description
Purpose	Application area, motivation, goals, requirements, and preconditions under which the system operates
Scope	For the intended system, define the higher-level overarching system of which it is a component, its own lower-level component systems, and relationship to peer systems
Technology	Underlying technology that enables the System and is used by it, level of technology maturity, evolutionary or disruptive
Economics	Business case for the System, Revenues, Cost structure, who pays, who profits
Politics	Regulatory considerations, public funding, benefits, legislative support, popular support, volatility of support
Structure	Identification of higher-level System, interfaces to and interaction with sibling Systems, process structure, precursor to System design

1.2.2 The Information System Structure

A central thesis of the first IPA report is that any Information System can be described technically by the components shown in Figure 1, described in the following.

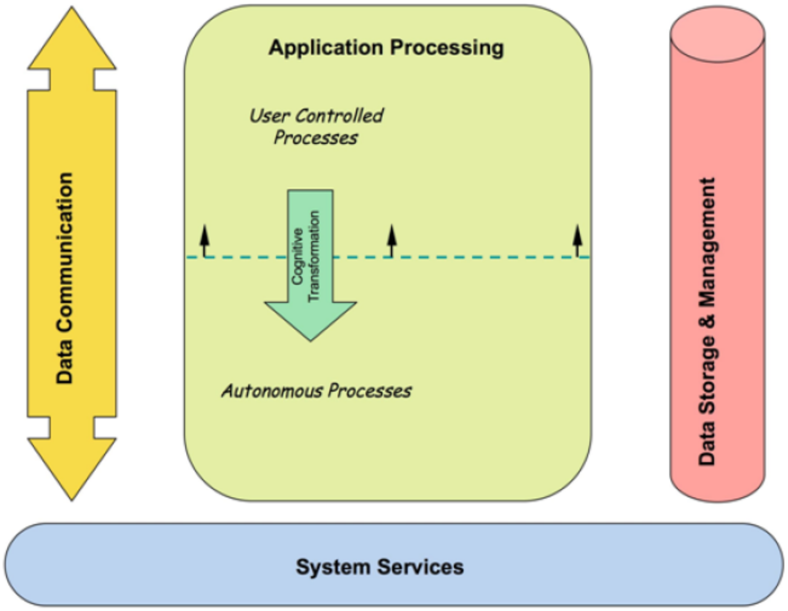


Figure 1: Information System Structure. From [IPA_10]

- **System Services** – an abstract representation of underlying hardware and low-level support services, System Services are the functions and processes provided by operating systems and other system functions to support system operation.
- **Data Storage and Management** – the processes and media associated with the replication, storage, and retrieval of the data used by the information processing system.
- **Data Communications** – the processes and systems used to convey data from one information system to another, generally via data replication
- **Application Processing** – application dependent, these processes define the basic functionality of the system and use and transform information by combining data from multiple sources, computing to form new data, and delivering information in a form that advances the application's objectives. It is assumed that these processes can be user controlled or autonomous and that cognitive radio is representative of the increasingly autonomous implementation of these processes for wireless information process systems

1.2.3 Information System Transaction Cycle and Modeling Components

The IPA extended the structure to consider the transfer of information from an originator to a recipient (possibly human, possibly autonomous processes) as shown in Figure 2 where processing combines data (stored or externally provided) with context to create information which is conveyed across a communications link with some context conveyed explicitly and some implicitly. The resulting information may be used to update the recipient's data and /or context for future communications.

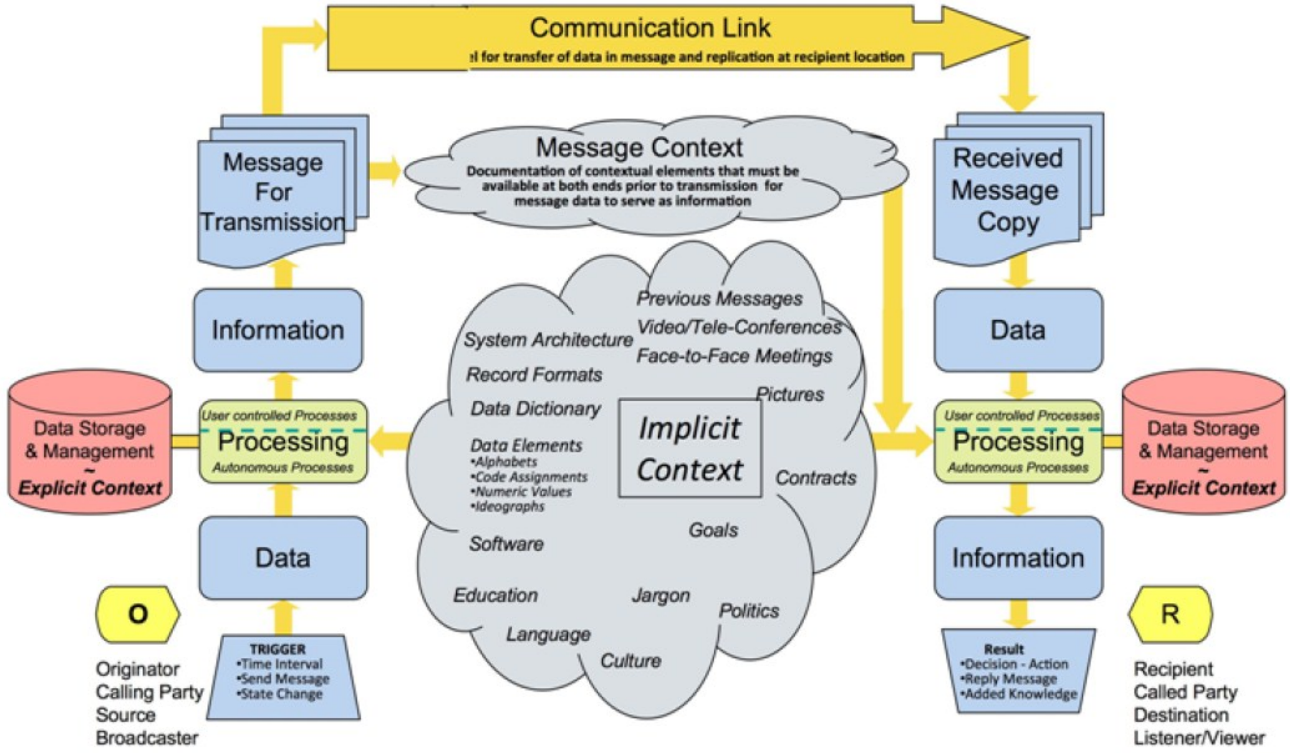


Figure 2: Information System Flow Model with Context. From [IPA_10]

1.3 Issues Addressed by this Document

This document analyzes a number of overarching issues that must be addressed when designing cognitive machine-to-machine systems. Some of these aspects involve the difference between communicating data and actionable information, communicating control and user information, and methods for efficient context transmission. Important issues are:

- Role of context in communications
- Context conveyance in a spectrum efficient manner
- Communication machine-to-machine with actionable information
- Insight into requirements for multiple information domains
- Elements of context that can be generalized for use in most communications applications
- Common methods for sharing information
- Importance of limiting scope to facilitate information exchange between disparate systems.
- Insights into design of extensible cognitive agents

1.4 Remaining Document Organization

The remainder of this document analyzes current communication systems information protocols and identifies characteristics of successful communications. A selection of various case studies involving machine-to-machine communications are examined including:

- Cursor on Target (CoT) – An application layer protocol for situational awareness for battlefield users communicating through disparate communications systems.
- Standard Spectrum Resource Format (SSRF) – A standard for sharing spectrum utilization, device capabilities, user requirements across a radio enterprise network.
- IEEE P1900.6 – A standard for sharing and coordinating spectrum sensing information and collection between TVBDs.
- Extensible Markup Language (XML) – An open source language developed for ease of human readability and machine interpretability used for the interchange of data between applications.
- Modeling Language for Mobility (MLM) – A standardized language with formal syntax and computer-processable semantics in which radios could express various aspects of communications, like their hardware and software capabilities.

These case studies are followed by a summary discussion of lessons that can be applied to cognitive radio systems leading to topics for future work needed to truly realize the promise of cognitive radios applied to more efficient spectrum utilization.

1.5 Introduction References

[FCC_02] Federal Communications Commission, "In the Matter of Additional Spectrum for Unlicensed Devices Below 900 MHz and in the 3 GHz Band." *ET Docket 02-380*. December 11, 2002.

[FCC_08] Federal Communications Commission, "In the Matter of Unlicensed Operation in the TV Broadcast Bands and Additional Spectrum for Unlicensed Devices Below 900 MHz and in the 3 GHz band. Second Report and Order and Memorandum Opinion and Order." *FCC 08-260*. November 4, 2008.

[IPA_10] Wireless Innovation Forum, "IPA - Information Process Architecture Volume I," *WINNF-09-P-0020-V1.0.0*, November 1, 2010.

[OET_08] Technical Research Branch Laboratory Division Office of Engineering and Technology, Federal Communications Commission, "Evaluation of the Performance of Prototype TV-Band White Space Devices Phase II," *OET Report FCC/OET 08-TR-1005*, Oct. 15, 2008.

2 Cursor on Target

Cursor on Target (CoT) is a U.S. Department of Defense (DoD) XML-based standard electronic portable data format to define location based data and to coordinate operation of equipment and personnel. Developed by Mitre in 2002 in support of the U.S. Air Force Electronic Systems Center (ESC), Mitre first demonstrated CoT "during a combined joint task force exercise in 2003, during which a Predator unmanned aircraft was able to operate (and) coordinate with manned aircraft." [Ucore_11] It has since grown to being used in more than 50 other CoT prototypes [Bryne_04] and is implemented by more than 70 nations. [Neuman_06]

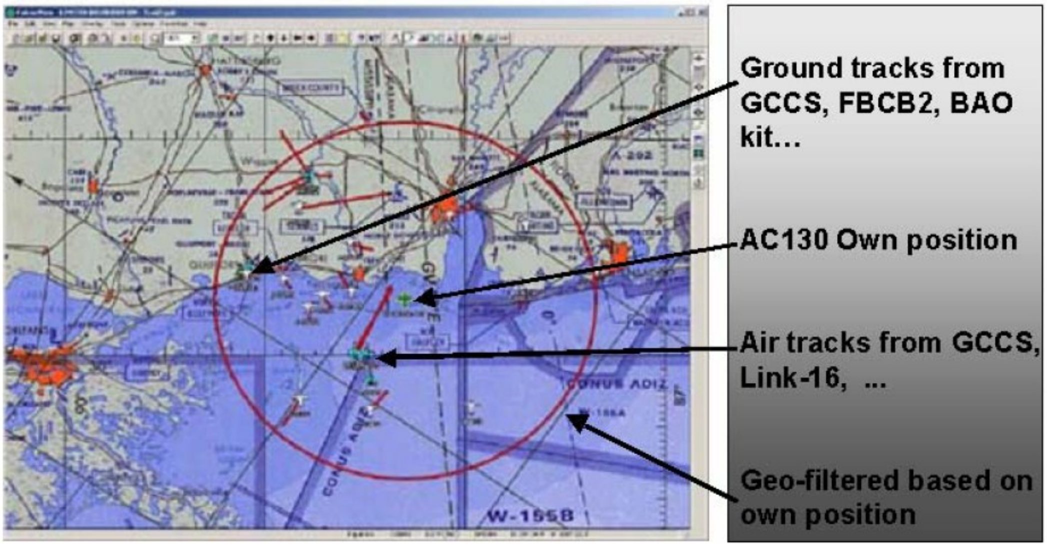


Figure 3: FalconView: One of the more popular viewers [Konstantopoulos_06]

CoT is commonly used for situational awareness, command and control, image processing, automated multi-asset management, airspace deconfliction, and weather information distribution [Robbins_07], to report and distribute sensor and Unmanned Aerial Vehicle (UAV) information, and to send tasking requests to individual or groups of objects in the CoT network (e.g., reposition a UAV or request new imagery from a camera). It provides a common context within which disparate user groups and situational awareness systems can communicate. Other uses include:

- "overlaying special ops targets, Army blue force positions, (*Air Force*) air situational awareness, and the joint Common Operating Picture all onto one display" [Schaeffer_05]

- "a MITRE team fused target information from a laser range finder, a compass, and a GPS receiver and then sent the data to an intelligence system to be refined for high-precision resolution. From there the data was relayed over a Link 16 radio to an F15E jet fighter to be automatically downloaded to onboard precision-guided munitions." [Byrne_04]

It accomplishes this via a common protocol and conventions that build on commonly available tools.

We studied CoT in this effort and report on it in this section because of CoT's success in facilitating communication between distinct user groups, its support for machine-to-machine (M2M) communications and automated operation, and the way that it exploits shared context to reduce bandwidth usage.

2.1 System Objectives

A long-standing problem with military communications is voice communications over noisy channels which can lead to mis-interpretation. long transmission times and can degrade mission effectiveness. Along with other solutions such as the phonetic alphabet (e.g., Alpha, Bravo, Charlie), Cursor on Target tries to solve this via a protocol appropriate for M2M communications over lossy channels as emphasized in the following popular origin story for CoT. [Schaeffer_05]

> "In April 2002 at the C2ISR Summit, Gen John Jumper (Chief of Staff for the Air Force) gave an impassioned plea to find ways to horizontally integrate machines directly talking to other machines to eliminate time-consuming and error-prone human translations. His "Sergeant Matt" story described a special ops warrior riding a donkey, laser designating targets using a handful of non-integrated machines, and manually performing calculations that ended with long voice transmissions over noisy radios, epitomizing the current state of warfare. What Jumper envisioned was machine-to-machine (M2M) automation that would achieve his vision that "the sum of all wisdom is a cursor over the target."

While equating CoT to "the sum of all wisdom" is a bit of hyperbole, CoT does distill vast amounts of data into actionable and easily understood information (within its context). This then allows [Byrne_04]

> "battle commanders to be able to mouse over an aerial view of enemy positions, point, click a cursor, and watch as the target is eliminated. Cursor on Target provides real-time access to secure and reliable information."

CoT critically provides and conveys context so that the aforementioned battle commanders know if they are eliminating an enemy target or protecting a friendly unit.

Rather than trying to address all possible types of communication, CoT focuses on a particular set of commonly needed information that addresses the following three questions about objects and events on the battlefield. [Konstantopoulos_06]

- What is it? Is it a friend or foe? Is it a tank, a UAV, a sensor or an anticipated event (e.g., weather front)?
- Where is it? What is its location for targeting and for coordinating movements? What is the uncertainty in the location?
- When is it? For what period of time is this information valid or when will it be valid (e.g., for coordinating movements)?

The communication of this information is required to be reliably communicated with minimal bandwidth to allow operation over a large number of different networks.

2.2 System Design

CoT achieves its objectives via a terse XML-schema and various messaging conventions. Because of its design, only a few thousand lines of code are required to implement [Neuman_06]

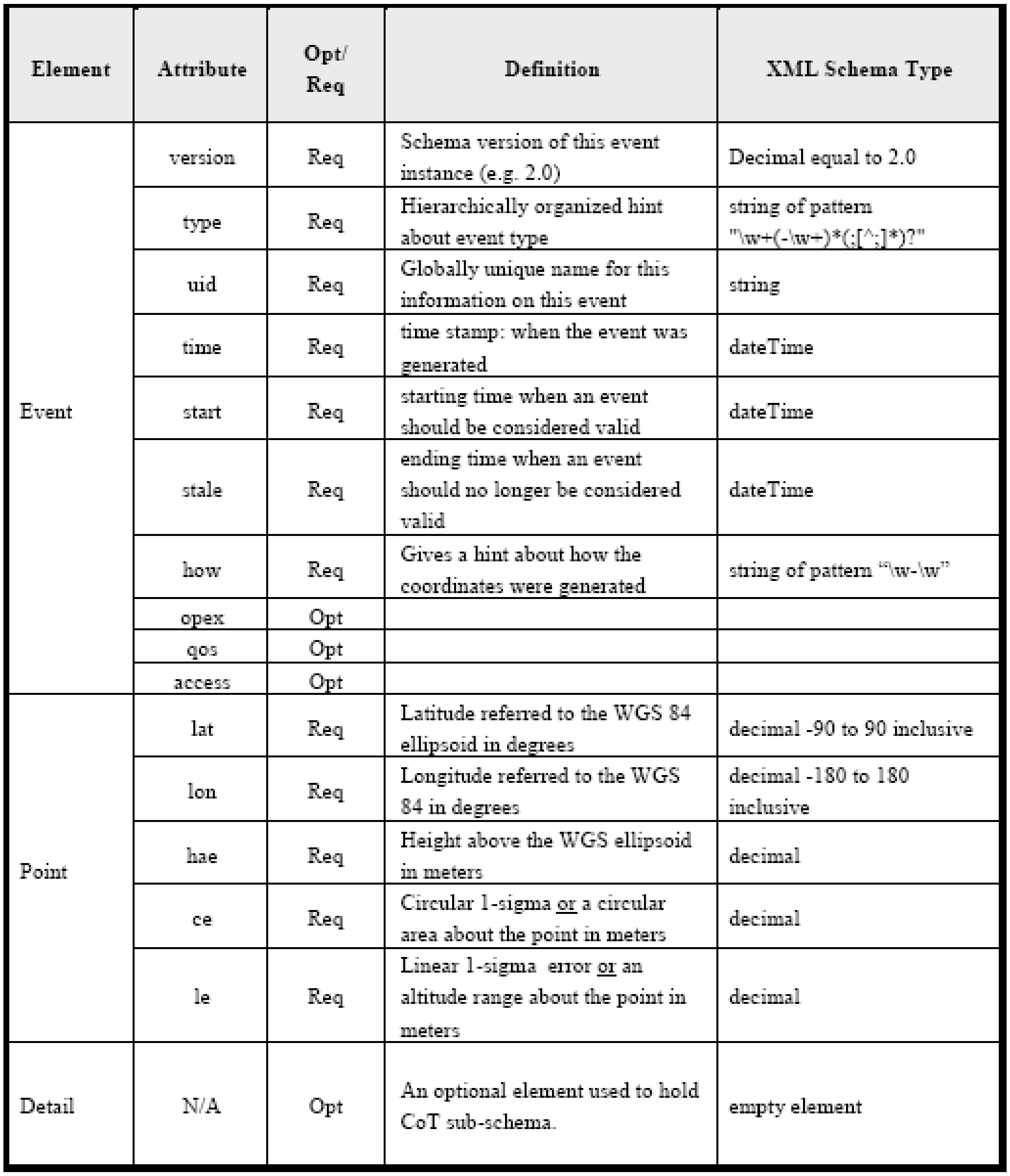
2.2.1 Architecture

Broadly, a CoT system is implemented over a network that connects a server with multiple edge devices (clients) through intermediate routers. The edge devices may be pure consumers of information provided by the server, may be producers of information, or both. Both information and requests (and replies) flow over the network, via TCP/IP messages. CoT supports both push and pull methods due to relative benefits of both approaches [Konstantopoulos_06]

> Theoretically speaking, push models are more adapted to transmitting Battlespace awareness/ Common Operating Pictures (COP) information, since the server retains the knowledge of when important data has changed in order to initiate a new transmission with clients.
>
> …
>
> In practice, clients are the final authority on which data is important (to them), and thus it is ultimately better to leave it up to the clients to initiate a pull for new data rather than having the server force-feeding them data that may be fresh but of marginal consequence. Moreover, clients may fuse data from multiple servers, and fusing is easier when the client is in control of data refresh activity.

2.2.2 CoT XML Schema

The base XML schema for CoT is shown in Figure 4. XML has following advantages as an implementation language, including wide availability of commercially available tools for processing XML, extensibility, and capability to be both machine and human readable. In XML format, information is formatted as a sequence of strings, rather than in a record with predefined fields. Such a format is less efficient, but does aid in human readability and extensibility to additional platforms.



Element	Attribute	Opt/ Req	Definition	XML Schema Type
Event	version	Req	Schema version of this event instance (e.g. 2.0)	Decimal equal to 2.0
	type	Req	Hierarchically organized hint about event type	string of pattern "\w+(-\w+)*(;[^;]*)?"
	uid	Req	Globally unique name for this information on this event	string
	time	Req	time stamp: when the event was generated	dateTime
	start	Req	starting time when an event should be considered valid	dateTime
	stale	Req	ending time when an event should no longer be considered valid	dateTime
	how	Req	Gives a hint about how the coordinates were generated	string of pattern "\w-\w"
	opex	Opt		
	qos	Opt		
	access	Opt		
Point	lat	Req	Latitude referred to the WGS 84 ellipsoid in degrees	decimal -90 to 90 inclusive
	lon	Req	Longitude referred to the WGS 84 in degrees	decimal -180 to 180 inclusive
	hae	Req	Height above the WGS ellipsoid in meters	decimal
	ce	Req	Circular 1-sigma or a circular area about the point in meters	decimal
	le	Req	Linear 1-sigma error or an altitude range about the point in meters	decimal
Detail	N/A	Opt	An optional element used to hold CoT sub-schema.	empty element

Figure 4: CoT XML Schema Base Schema [Kristan_09].

There are three primary elements in a CoT message – *Event*, *Point*, and *Detail*.

2.2.2.1 Event

The *Event* element answers two of the three questions: What and When. What is addressed by the type field, which is a hierarchical descriptor that can provide a detailed description of what is being reported in this message, e.g., what type of tank is seen, what domain is it operating in and its affiliation. This has been described as "Organized much like an object hierarchy used in object-oriented programming." [Konstantopoulos_06]

The *type* field is a hierarchical classification of the object of the message with symbols drawn from a shared dictionary. A partial breakdown of the hierarchy is shown in Figure 5 for messages about “atoms” (physical things), which are successively classified by affiliation (e.g., friend-or-foe), domain (land, sea, or air), and then more detailed classifiers. For example, a tank might be described in the hierarchy as thing (atom) -> Friendly (affiliation)- > Ground (domain) ->tank -> what type of tank-> … These classifications follow the MIL-STD-2525[1] as much as possible, though CoT specific classifications are also sometimes necessary. To differentiate between what is a CoT specific classification and a 2525 code, labels from 2525 are written in upper case, while labels from CoT are in lower case. [Konstantopoulos_06]

$$\text{ValidAtomicEventCombinatins} =$$

$$a\,(\text{atom}) - \left\{ \begin{array}{c} p\,(\text{pending}) \\ u\,(\text{unknown}) \\ a\,(\text{assumedfriend}) \\ f\,(\text{friend}) \\ n\,(\text{neutral}) \\ s\,(\text{suspect}) \\ h\,(\text{hostile}) \\ j\,(\text{joker}) \\ k\,(\text{faker}) \\ o\,(\text{nonespecified}) \\ x\,(\text{other}) \end{array} \right. - \left\{ \begin{array}{c} P\,(\text{Space}) \\ A\,(\text{Air}) \\ G\,\text{Ground)} \\ S\,(\text{Sea Surface}) \\ U\,(\text{Sea Subsurface}) \\ X\,(\text{Other}) \end{array} \right. - [\text{MIL-STD-2525FunctionCode(s)}]$$

Figure 5: Example type hierarchies [Kristan_09]

The use of a hierarchical type field has the following implications.

1) Only a handful of classifications are possible at each level of the hierarchy, which means that a single letter can be used to convey the relevant information at each level, which reduces bandwidth usage.
2) Less capable devices or users only interested in information defined at higher levels need only parse a part of the type description.
3) The information assumed to be the most relevant to the most users is placed first to hasten processing.

The *Event* field also answers the When question, specifically addressing the start time at which the information is valid and the stale time at which the information should no longer be considered valid. Note that to aid human readability, time is specified in a year-date-time format rather than a decimal format. Additional context is provided in the Event field to aid in processing including:

- Determining what version of schema is being used. In theory, this should simplify compatibility issues, though the event schema has been stable at 2.0 for some time.

---

[1] See http://www.mapsymbs.com/ms2525b_ch1_full.pdf

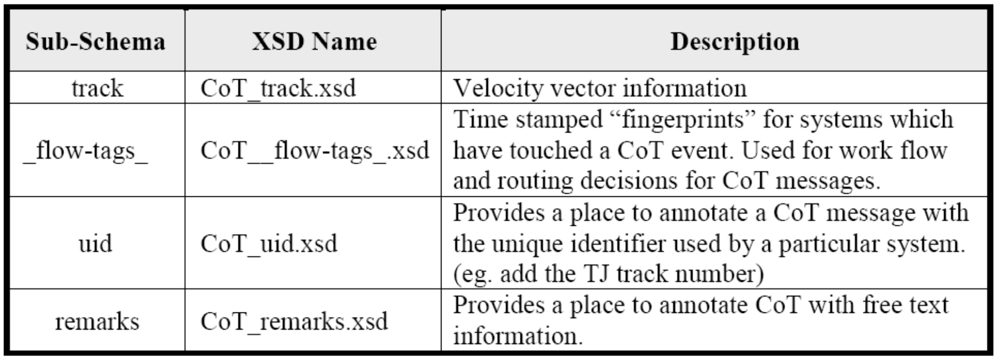
- A unique identifier about the event (not the sender!) which facilitates information fusion from multiple sources.
- A note about how the information was generated (e.g., manually or by machine) so the recipient of the information can consider this in weighting the quality of the reported information.

2.2.2.2 Point

The *Point* field answers the Where question. The *Point* field specifies where the event is located in terms of geographic location and height (e.g., on the ground or in the air). Additionally, an uncertainty in the location is also conveyed in terms of height and radius, thereby defining a cylinder of location uncertainty. Height is measured as “height above ellipsoid” based on WGS-84 ellipsoid, the same convention used with GPS. [Konstantopoulos_06]

2.2.2.3 Detail

The Detail field is optional and included to allow the use of sub-schema to communicate additional information beyond the base schema, e.g., requests for actions or very specific language such as weather information. By allowing this custom sub-schema definition while preserving the main schema, more specialized systems or user groups can still inter-operate with and share battlefield information with more limited or generalized systems. Examples of sub-schema are shown below. The Details field is also used to link together multiple UIDs to express relationships between the UIDs, e.g., a group of objects or that one object should perform some function on another object. [Konstantopoulos_06]

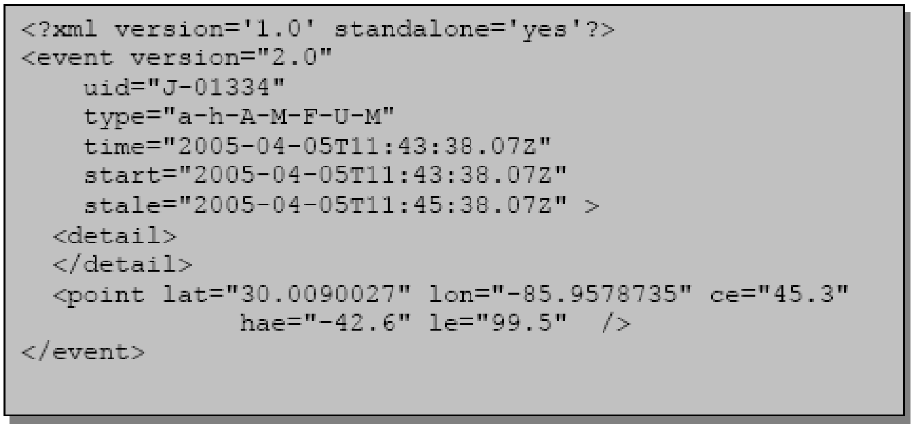


Sub-Schema	XSD Name	Description
track	CoT_track.xsd	Velocity vector information
flow-tags	CoT__flow-tags_.xsd	Time stamped “fingerprints” for systems which have touched a CoT event. Used for work flow and routing decisions for CoT messages.
uid	CoT_uid.xsd	Provides a place to annotate a CoT message with the unique identifier used by a particular system. (eg. add the TJ track number)
remarks	CoT_remarks.xsd	Provides a place to annotate CoT with free text information.

Figure 6: Example subschema. From [Kristan_09]

2.2.2.4 Sample Message

A sample CoT message is shown below, which can be partially expanded as an atom (a, i.e., a thing) hostile (h) in the Air (A) aircraft (M) fixed wing (F) was spotted at 11:43 (Zulu) on April 5, 2005. The aircraft has been assigned the identifier “J-01334” for later processing and was at latitude 30.090027.



```
<?xml version='1.0' standalone='yes'?>
<event version="2.0"
    uid="J-01334"
    type="a-h-A-M-F-U-M"
    time="2005-04-05T11:43:38.07Z"
    start="2005-04-05T11:43:38.07Z"
    stale="2005-04-05T11:45:38.07Z" >
  <detail>
  </detail>
  <point lat="30.0090027" lon="-85.9578735" ce="45.3"
              hae="-42.6" le="99.5"  />
</event>
```

Figure 7: Example Cot message reporting a hostile fixed wing air craft at the latitude and longitude specified, but only valid at the time of reporting. From [Kristan_09].

2.2.3 CoT Conventions and Functions

In addition to the overt information transfer in a CoT message, there are a number of conventions that help processing and reduce bandwidth by creating a shared context. An example is accounting for the time validity of a message as described in [Konstantopoulos_06].

> "Any geolocalized point can thus be considered "valid" and constitutes actionable information until the current time & date overtakes the time and date identified in the *stale* sub-element. In this way, DoD systems need not be constantly exchanging WWW information in order to keep their Common Operating Picture (COP) synchronized at all times.
>
> …
>
> In other words, what is really transmitted between systems sharing WWW information consists of "diffgrams", or "deltas" which are relative to changed data."

CoT also supports the definition of tests or subscriptions that allows an end-client to limit how much data is streamed to it. In a fairly expressive manner, a client can request that the server or router limit the stream to any subclass defined by any part of the message, from type to Unique Identifier (UID) to location to messages within a certain time frame to affiliations. Because of the hierarchy, this is a relatively simple parsing activity for the server.

2.3 Lessons Learned from CoT for IPA

From our review of CoT, we are able to glean the following insights.

- **Limiting scope and enforcing a common language facilitates communications across disparate groups**
  CoT only addresses three fundamental questions – what, where, and when – and then greatly constrains what types of items are communicated. This places a much smaller burden on the disparate user communities for interfacing with the system and each other. Extensions beyond the base (or common) level of communications are possible by extending the details field.

- **Efficient coding can be derived from a limited dictionary (terse schemas)**
  Shorter words (generally one letter) can be used to convey meaning because of the limited dictionary. Further, the dictionary is segmented so that for each field there are only a limited number of words that can be used. This greatly simplifies machine-to-machine communications, which is at the heart of CoT, and facilitates the integration into different nations' systems.

- **Implicit context (conventions) related to time reduce bandwidth requirements**
  Many fewer messages can be sent because of a couple conventions related to timing (updates are only sent when information changes or when information becomes stale). Loosely, this is akin to the savings achievable with event-based communications or computation.

- **Responsibilities for different types of information are segmented and fulfilled by nodes in different roles**
  In CoT, synthesized information (e.g., all sensor info within a specified area) is supplied by a server. Individual nodes, however, are responsible for keeping the network apprised of their unique information via the server.

- **Strong hierarchies can help communications be both extensible and efficient**
  The key to making this happen was an ability to discover other device's capabilities based on type (from which many capabilities can be inferred) rather than having to directly transfer lists of capabilities.

- **Development time is reduced by leveraging existing tools**
  Rather than custom designing an entire new protocol stack, CoT is ultimately an application layer protocol that sits on top of many other protocols, e.g., XML and TCP/IP.

- **Marketing to humans matters for technology adoption**
  There are many other systems available for sharing situational awareness data that could have emerged as a de-facto military standard, though perhaps none with a name as catchy as "Cursor on Target". At the same time, if CoT had a different name, it might realize wider adoption beyond the military.

2.4 Cursor on Target References

[Byrne_04] R. Byrne, "'Cursor on Target'" Improves Efficiency" The Edge, MITRE, Fall 2004. Available online: http://www.mitre.org/news/the_edge/fall_04/edge_fall_04.pdf

[Konstantopoulos_06] D. Konstantopoulos, J. Johnston, "Data Schemas for Net-Centric Situational Awareness," CCRTS 2006, Available online: http://www.dodccrp.org/events/2006_CCRTS/html/papers/073.pdf

[Kristan_09] M. Kristan, J. Hamalainen, D. Robbins, P. Newell, "Cursor on Target Message Router User's Guide," MITRE PRODUCT MP090284, November 2009. Available online: http://www.mitre.org/work/tech_papers/2010/09_4937/09_4937.pdf

[Neuman_06] T. Neuman, "Cursor on Target (CoT): The Future of Network Centric Warfare," http://www.angelfire.com/planet/gb3020_cursor_on_tgt/index.html, 2006.

[Robbins_07] D. Robbins, "Unmanned Aircraft Operational Integration Using MITRE's Cursor on Target," The EDGE, Summer 2007, vol. 10, no 2. Available online: http://www.mitre.org/news/the_edge/summer_07/edge_summer_07.pdf

[Schaeffer_05] K. Schaeffer, T. Gibbons, Jr., "Enhancing the Extended Awareness Capability of the ESG: Integrating Shotspotter and Cursor on Target Technologies with Unmanned Aerial Vehicles to Enhance the Mission Capability of the ESG," Masters Thesis, Naval Post Graduate School, June 2005. Available online: http://edocs.nps.edu/npspubs/scholarly/theses/2005/Jun/05Jun_Schaeffer.pdf

[Ucore_11] "Ucore", Wikipedia. Retrieved 4/29/11. http://en.wikipedia.org/wiki/UCore

3 Standard Spectrum Resource Format

As written in [MCEB_09]:

> Standard Spectrum Resource Format (SSRF) is a format for exchanging data related to spectrum management within the Department of Defense (DOD). SSRF-compliant systems will be able to exchange electromagnetic spectrum data with the National Telecommunications and Information Administration (NTIA), the North Atlantic Treaty Organization (NATO), and with Combined Communications-Electronics Board (CCEB) nations. SSRF enables the development of tools to more efficiently manage a finite resource that is in increasing demand by the warfighter and is key to DOD's Net-Centric Data Strategy. SSRF may be used within and between organizations, between differing systems that require access to spectrum management data and, potentially, with sophisticated network-enabled emitters.

The following briefly describes the motivation and history for developing SSRF.

The US Department of Defense (DoD) must manage numerous different wireless communications and electronic warfare systems that are deployed all over the world and are arrayed across the spectrum, as illustrated in Figure 8. Spectrum for each of these systems must be coordinated and deconflicted with other nations, managed, and access prioritized.

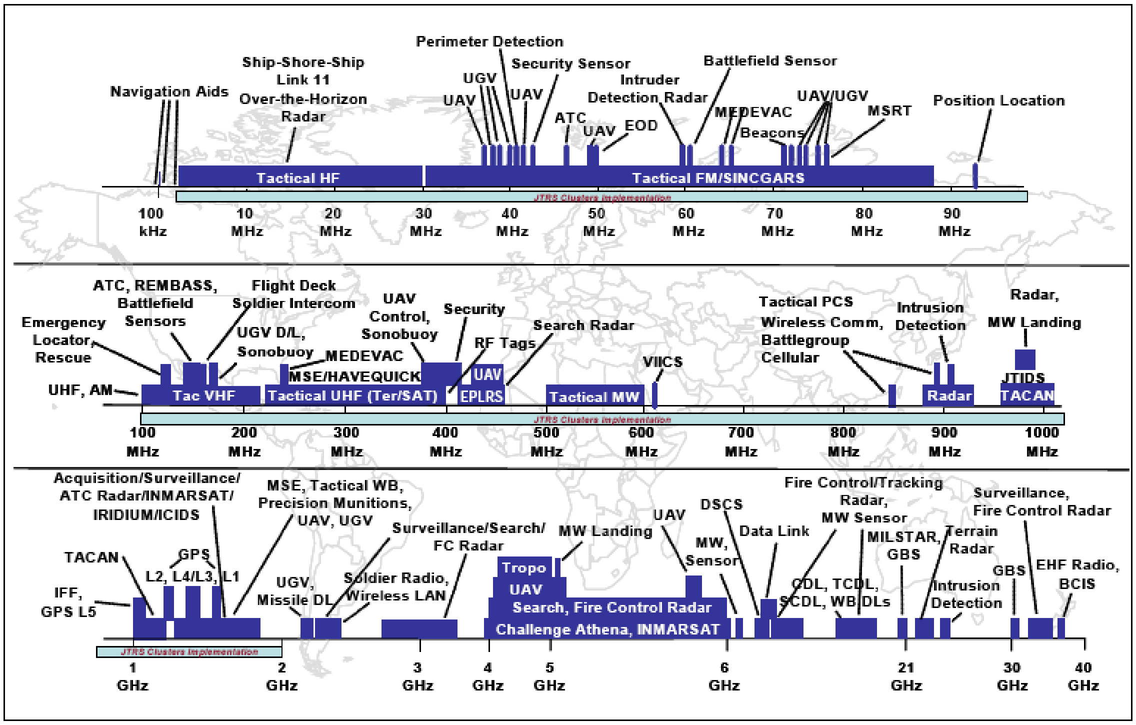


Figure 8: Warfighter Spectrum Use Below 40 GHz From: http://dodreports.com/pdf/ada476476.pdf

Further complicating this task is the wide variety of existing spectrum management tools and data representations, including [MCEB_09]:

- **SFAF**: The Standard Frequency Action Format (SFAF) is a line-oriented text format used by DOD, and by U.S. allies and coalition partners who use SPECTRUM XXI.
- **GMF Card**: The Government Master File (GMF) Card is a line-oriented text format used by NTIA for frequency assignment data.
- **14 point format**: 14 Point is a line-oriented text message format used to exchange frequency assignment data in Partnership for Peace (PFP) Nations and some NATO Nations.
- **SMADEF**: The original line-oriented non-XML format used by NATO for both frequency assignment and spectrum supportability data.
- **DD Form 1494**: Paper form used for spectrum supportability by the U.S., both internally and with many allies.
- **SCS Files**: Spectrum Certification System files, the electronic equivalent of DD Form 1494, have been used within DOD and with NTIA to exchange spectrum supportability data.
- **Forms 33, 34, and 35**: Paper forms used by NTIA to collect, process, and distribute spectrum supportability data.
- **EL-CID Files**: Equipment Location - Certification Information Database files are ZIP archives of XML data and binary attachments used by NTIA and federal agencies to exchange Spectrum Supportability data. EL-CID files were the first step toward an XML format for supportability data.

Further, as noted in [Anderson_07], many different entities must coordinate with one another for effective spectrum management:

> the [electromagnetic spectrum] is heavily "occupied" by military systems in a Joint Operations Area (JOA). Each Geographic Combat Commander (GCC) is tasked by Chairman of the Joint Chiefs of Staff Instruction (CJCSI) 3320.01B to "establish a standing frequency management structure that includes a Joint Frequency Management Office (JFMO)."[68] At the Joint Task Force (JTF) level, spectrum may be managed by a JTF Spectrum Management Element (JSME).[69] Additionally, the JTF may establish an Electronic Warfare Coordination Cell (EWCC) to support EW planning and policies in the JOA.[70] The primary tool used to manage spectrum in the JOA is the Joint Restricted Frequency List (JFRL), which lists the networks and frequencies deemed critical to JTF objective.[71]

To reduce incompatibilities and facilitate coordination among the many different spectrum management entities: [MCEB_09]

> "NATO Frequency Management Subcommittee (FMSC) chartered a working group to develop [Spectrum Management Allied Data Exchange Format-eXtensible Markup Language] as the key to interoperability between spectrum management organizations in all NATO Nations, NATO Commands and other Nations (such as PFP Nations). The approach followed in the development of SMADEF-XML was to create a standard which could satisfy all the needs of the spectrum managers, at the national and international levels, and at all levels of the hierarchy from Ministry of Defense (MoD) and NATO HQ down to the Force Elements. The result is a harmonized multi-purpose interface which can support all the spectrum management business processes: frequency assignment, spectrum supportability, JRFL dissemination, interference reports, etc, as well as providing a common way to capture and manipulate frequency management information to improve these processes.
>
> In order to ensure support for Warfighter requirements and interoperability with NTIA, DOD maintains SSRF as a separate entity, based on SMADEF-XML. For similar reasons, NTIA maintains Office of Spectrum Management Data Dictionary (OSMDD) as its own implementation of the standard."

In [DoD_11], the Department of Defense stated its intention to stand up a central spectrum data administrator – the DoD Spectrum Data Administrator (DSDA), which as its first task shall:

> Ensure that the SSRF is registered as the authoritative data standard for spectrum-related data in the DoD Information Technology Standards Registry (DISR).

To better understand how to facilitate interoperability between disparate user groups in the context of facilitating actionable machine-to-machine communications, this section reviews SSRF's objectives, architecture, and implications for the IPA.

3.1 System Objectives

SSRF has the primary objective of facilitating the coordination of spectrum operations between commands, agencies, and allies by standardizing data elements and representations. This in turn aids in the coordination of the use of spectrum, aids in the development of software that supports spectrum managers, and has more tangible impacts such as reducing spectrum fratricide. At a macro-level, specific entities targeted for interoperability include to NTIA, NATO, and Combined Communications-Electronics Board (CCEB) nations. At a lower-level, SSRF also seeks to facilitate communications between a spectrum manager and many other spectrum "actors" as illustrated in Figure 9. Constraining the interoperability goal, SSRF also had to support requirements that information being exchanged may be at varying levels of classification.

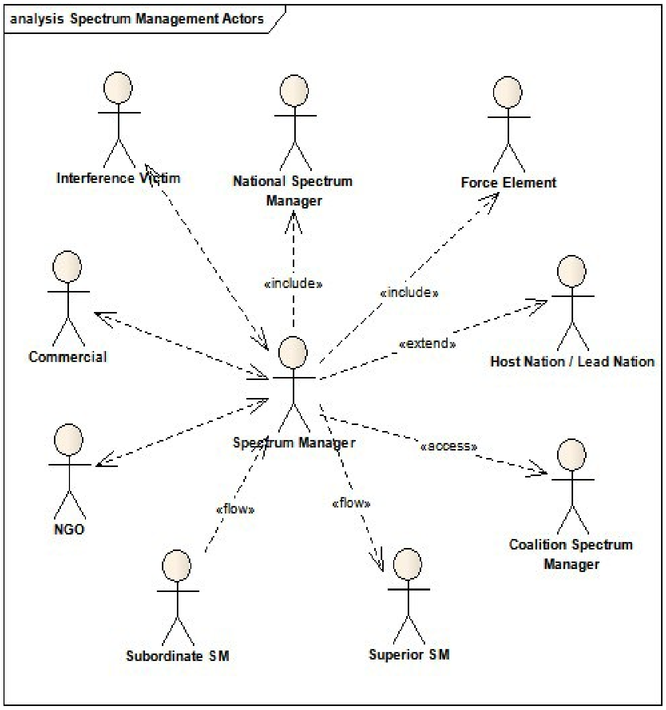


Figure 9: From Figure 1.4.3 in [MCEB_09]

To guide the development of SSRF, the set of tasks for a spectrum manager shown in Figure 10 were identified as core tasks.

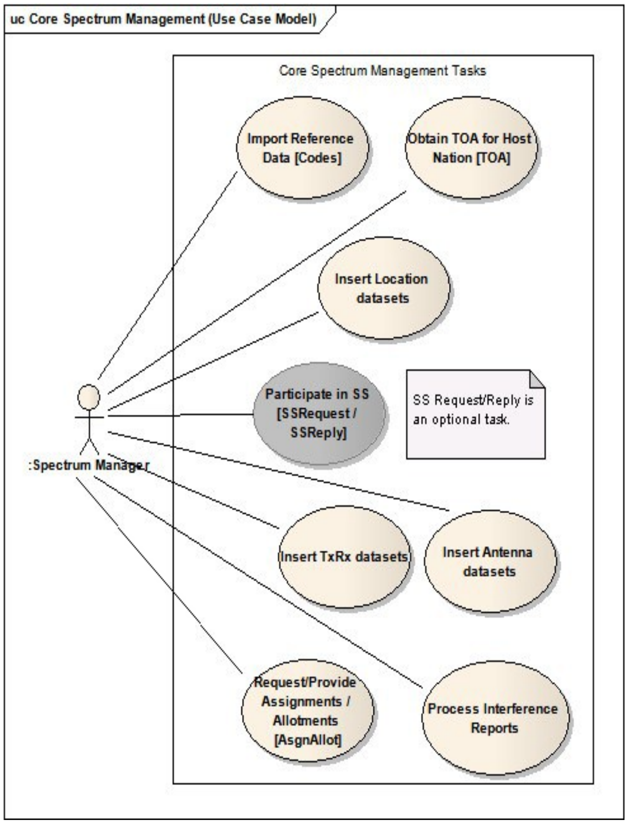


Figure 10: SSRF Manager Core Tasks. From Figure 2.1.1 in [MCEB_09]

3.2 System Design

3.2.1 Architecture

SSRF builds on XML as its base communications language for all communications between SSRF enabled tools.

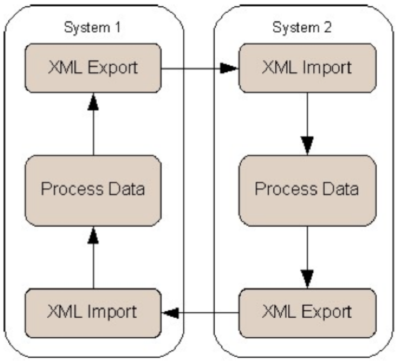


Figure 11: Basic Information Exchange between two SSRF-based tools. From Figure 1.4.1 in [MCEB_09].

The SSRF architecture assumes that there will be translation processes implemented to support data exchanges with legacy spectrum management tools that are not SSRF compliant, such as sending OSMDD formatted data to NTIA systems and SMADEF-XML to CCEB systems.

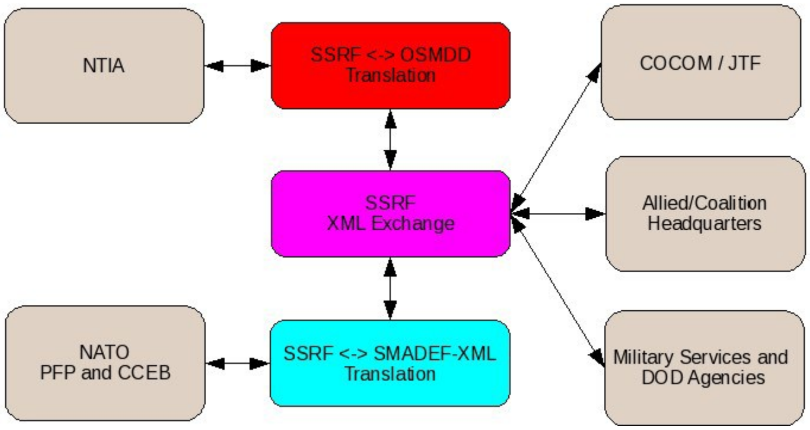
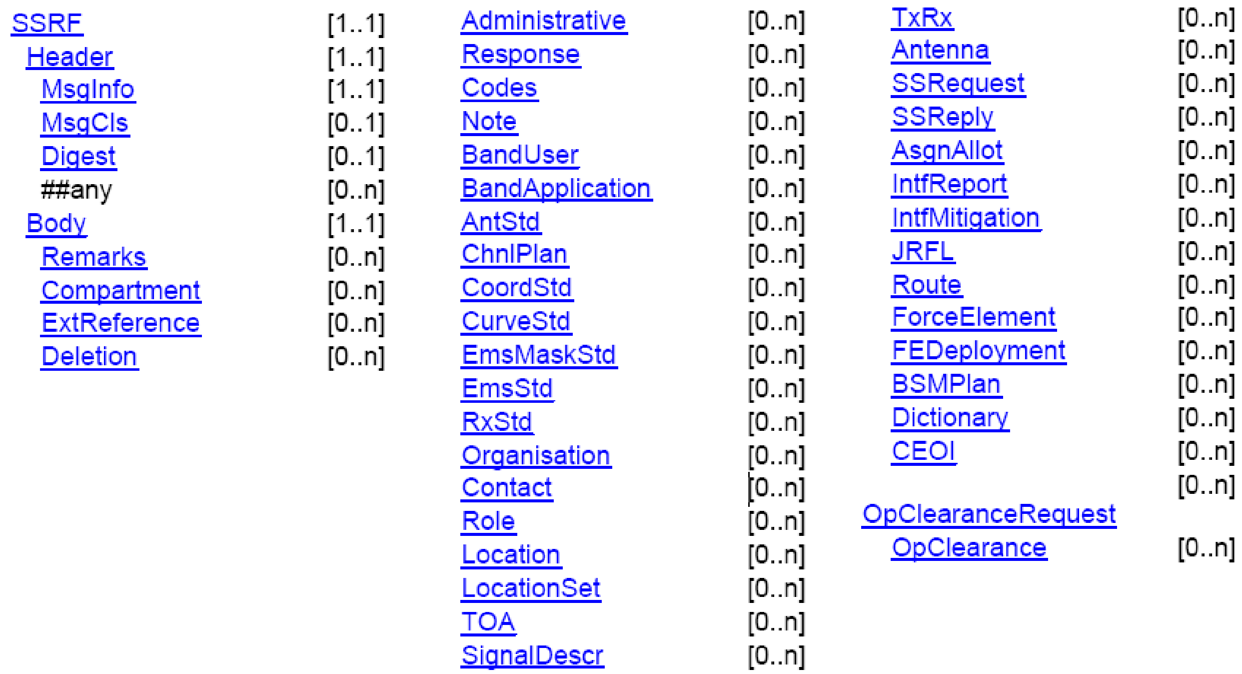


Figure 12: SSRF expects translation processes to be used to communicate with non-SSRF compliant systems while SSRF XML exchange is used elsewhere. From Figure 1.1.1 in [MCEB_09].

3.2.2 Basic Message Format

An SSRF message is an XML message with the following top-level fields. Many of these are optional (fields denoted with [0…]) and all can be expanded to provide more detail on a particular topic. Of note: the element <any> is a placeholder which allows national extensions providing additional protocol information, encryption, etc. When an extension is introduced, it must be defined in an XML Schema file (.xsd) and be made available to the recipients of the message. Note that having the appropriate .xsd file will allow the recipient to correctly parse the syntax of the SSRF message, but the recipient may still be in the dark as to the semantics of the element. The SSRF standard ([MCEB_09]) also defines a manual approval process for making changes to the SSRF messaging format.



SSRF	[1..1]
Header	[1..1]
MsgInfo	[1..1]
MsgCls	[0..1]
Digest	[0..1]
##any	[0..n]
Body	[1..1]
Remarks	[0..n]
Compartment	[0..n]
ExtReference	[0..n]
Deletion	[0..n]

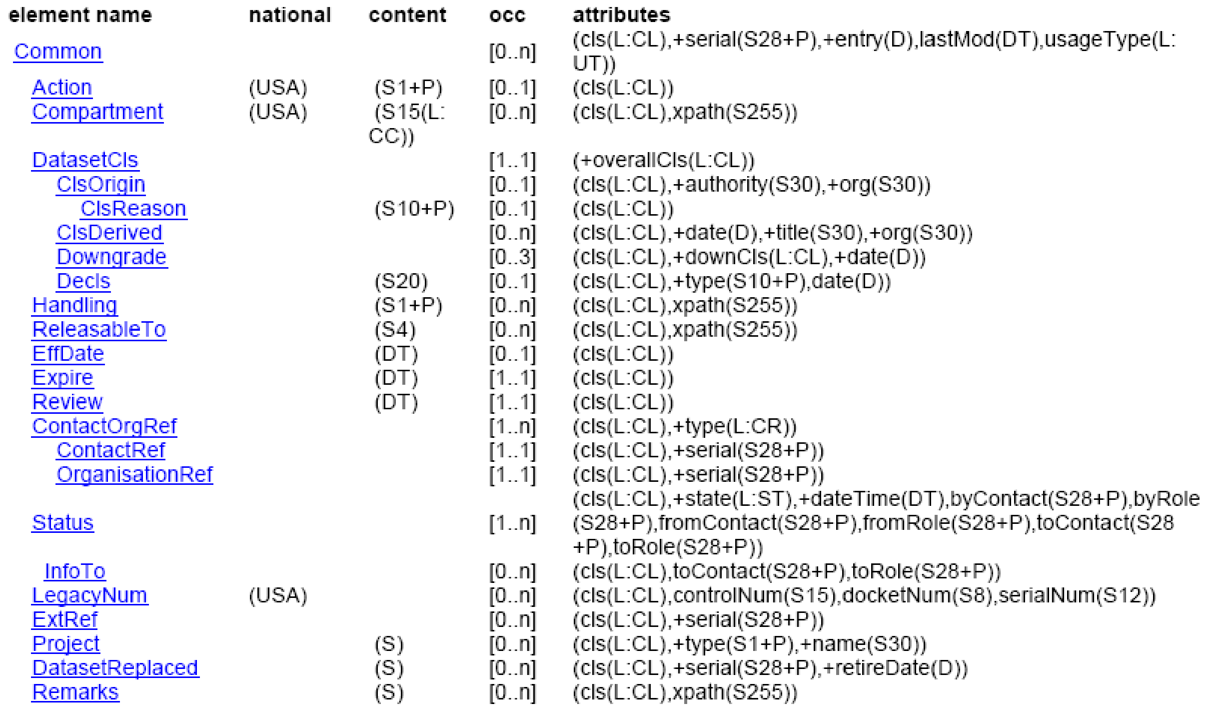
Administrative	[0..n]
Response	[0..n]
Codes	[0..n]
Note	[0..n]
BandUser	[0..n]
BandApplication	[0..n]
AntStd	[0..n]
ChnlPlan	[0..n]
CoordStd	[0..n]
CurveStd	[0..n]
EmsMaskStd	[0..n]
EmsStd	[0..n]
RxStd	[0..n]
Organisation	[0..n]
Contact	[0..n]
Role	[0..n]
Location	[0..n]
LocationSet	[0..n]
TOA	[0..n]
SignalDescr	[0..n]

TxRx	[0..n]
Antenna	[0..n]
SSRequest	[0..n]
SSReply	[0..n]
AsgnAllot	[0..n]
IntfReport	[0..n]
IntfMitigation	[0..n]
JRFL	[0..n]
Route	[0..n]
ForceElement	[0..n]
FEDeployment	[0..n]
BSMPlan	[0..n]
Dictionary	[0..n]
CEOI	[0..n]
	[0..n]
OpClearanceRequest	
OpClearance	[0..n]

Figure 13: Top Level Elements in SSRF. From Figure II-2 in [MCEB_09]

3.3 Common Element Inheritance

Each of these elements inherit attributes from the Common Element, which has the attributes shown below. By inheriting the attributes of the Common Element the SSRF design can then implement features such as providing unique time stamping to each element in a record and providing varying levels of classification to each element in a record.

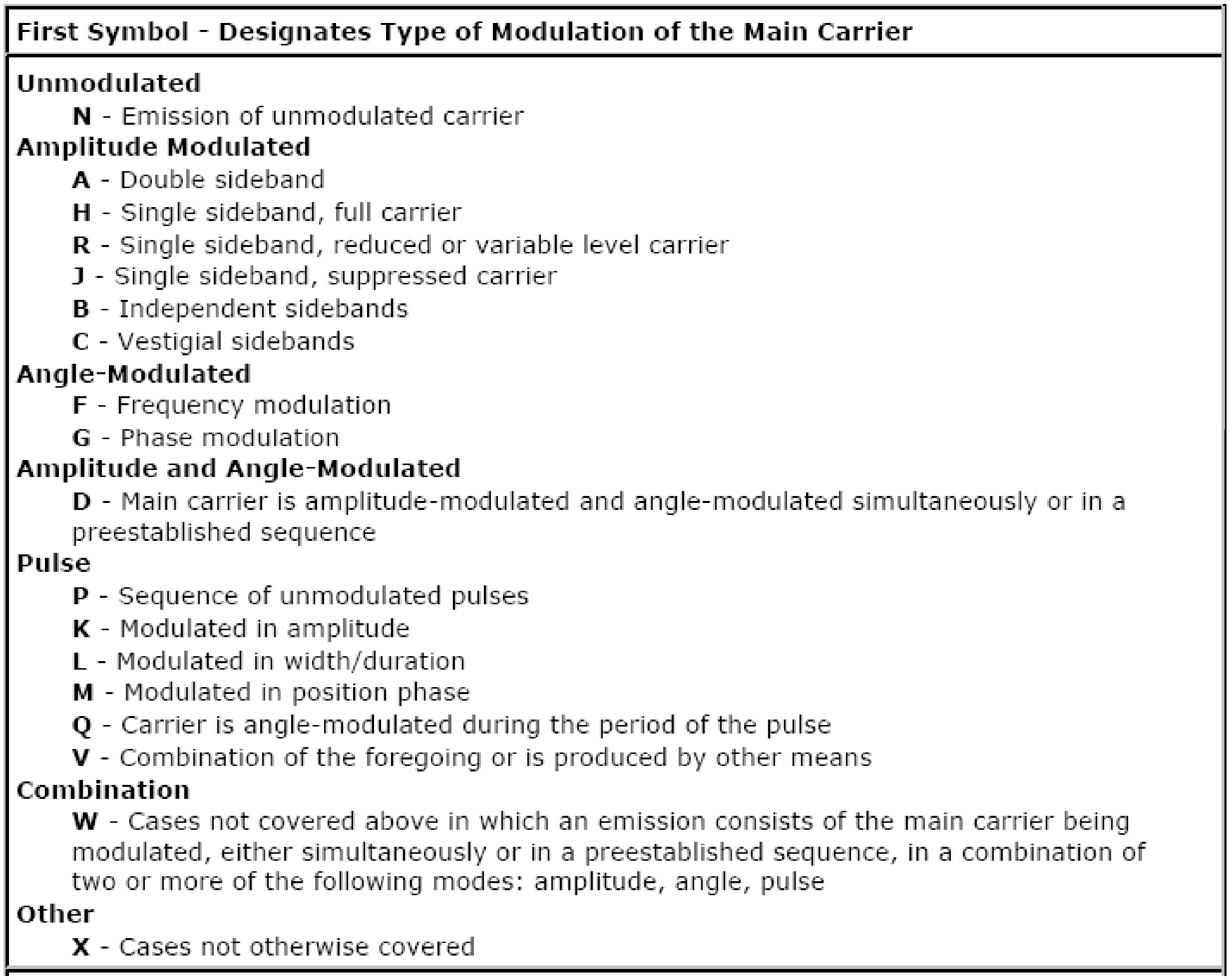


element name	national	content	occ	attributes
Common			[0..n]	(cls(L:CL),+serial(S28+P),+entry(D),lastMod(DT),usageType(L:UT))
Action	(USA)	(S1+P)	[0..1]	(cls(L:CL))
Compartment	(USA)	(S15(L:CC))	[0..n]	(cls(L:CL),xpath(S255))
DatasetCls			[1..1]	(+overallCls(L:CL))
ClsOrigin			[0..1]	(cls(L:CL),+authority(S30),+org(S30))
ClsReason		(S10+P)	[0..1]	(cls(L:CL))
ClsDerived			[0..n]	(cls(L:CL),+date(D),+title(S30),+org(S30))
Downgrade			[0..3]	(cls(L:CL),+downCls(L:CL),+date(D))
Decls		(S20)	[0..1]	(cls(L:CL),+type(S10+P),date(D))
Handling		(S1+P)	[0..n]	(cls(L:CL),xpath(S255))
ReleasableTo		(S4)	[0..n]	(cls(L:CL),xpath(S255))
EffDate		(DT)	[0..1]	(cls(L:CL))
Expire		(DT)	[1..1]	(cls(L:CL))
Review		(DT)	[1..1]	(cls(L:CL))
ContactOrgRef			[1..n]	(cls(L:CL),+type(L:CR))
ContactRef			[1..1]	(cls(L:CL),+serial(S28+P))
OrganisationRef			[1..1]	(cls(L:CL),+serial(S28+P))
Status			[1..n]	(cls(L:CL),+state(L:ST),+dateTime(DT),byContact(S28+P),byRole(S28+P),fromContact(S28+P),fromRole(S28+P),toContact(S28+P),toRole(S28+P))
InfoTo			[0..n]	(cls(L:CL),toContact(S28+P),toRole(S28+P))
LegacyNum	(USA)		[0..n]	(cls(L:CL),controlNum(S15),docketNum(S8),serialNum(S12))
ExtRef			[0..n]	(cls(L:CL),+serial(S28+P))
Project		(S)	[0..n]	(cls(L:CL),+type(S1+P),+name(S30))
DatasetReplaced		(S)	[0..n]	(cls(L:CL),+serial(S28+P),+retireDate(D))
Remarks		(S)	[0..n]	(cls(L:CL),xpath(S255))

Figure 14: Attributes of the Common Element in SSRF. From [MCEB_09]

3.4 Values

While the element names in SSRF remain largely human readable, e.g., "Note" or "AssgnAllot", non-numeric values are typically written in a more compact manner with symbols instead of writing out complete words. This saves on bandwidth (a little) while generally maintaining human readability. Like with CoT, sequences of symbols can be used to provide further classification. For instance the emsClass (emissions Class) attribute is used to define the kind of signal being transmitted. The emsClass code requires three symbols to be included in the data field with two optional fields with the allowable values for these symbols shown below.



First Symbol - Designates Type of Modulation of the Main Carrier

Unmodulated
- **N** - Emission of unmodulated carrier

Amplitude Modulated
- **A** - Double sideband
- **H** - Single sideband, full carrier
- **R** - Single sideband, reduced or variable level carrier
- **J** - Single sideband, suppressed carrier
- **B** - Independent sidebands
- **C** - Vestigial sidebands

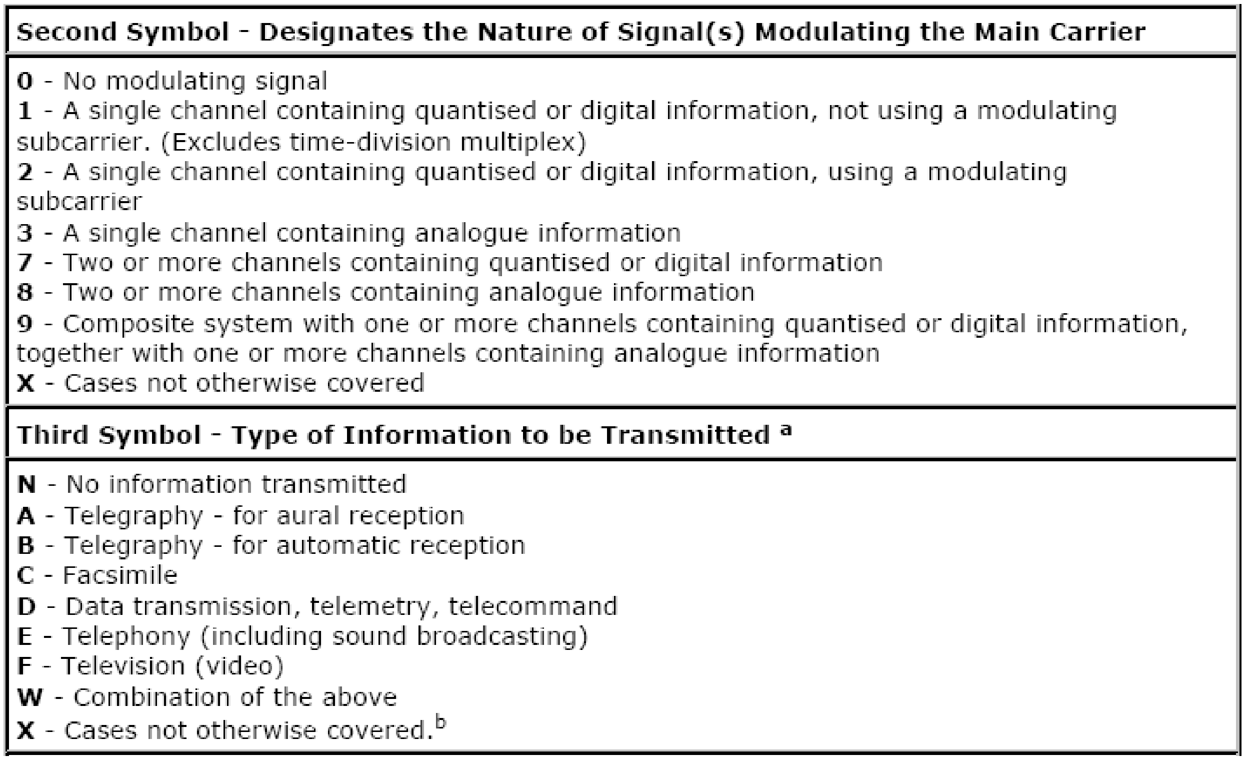
Angle-Modulated
- **F** - Frequency modulation
- **G** - Phase modulation

Amplitude and Angle-Modulated
- **D** - Main carrier is amplitude-modulated and angle-modulated simultaneously or in a preestablished sequence

Pulse
- **P** - Sequence of unmodulated pulses
- **K** - Modulated in amplitude
- **L** - Modulated in width/duration
- **M** - Modulated in position phase
- **Q** - Carrier is angle-modulated during the period of the pulse
- **V** - Combination of the foregoing or is produced by other means

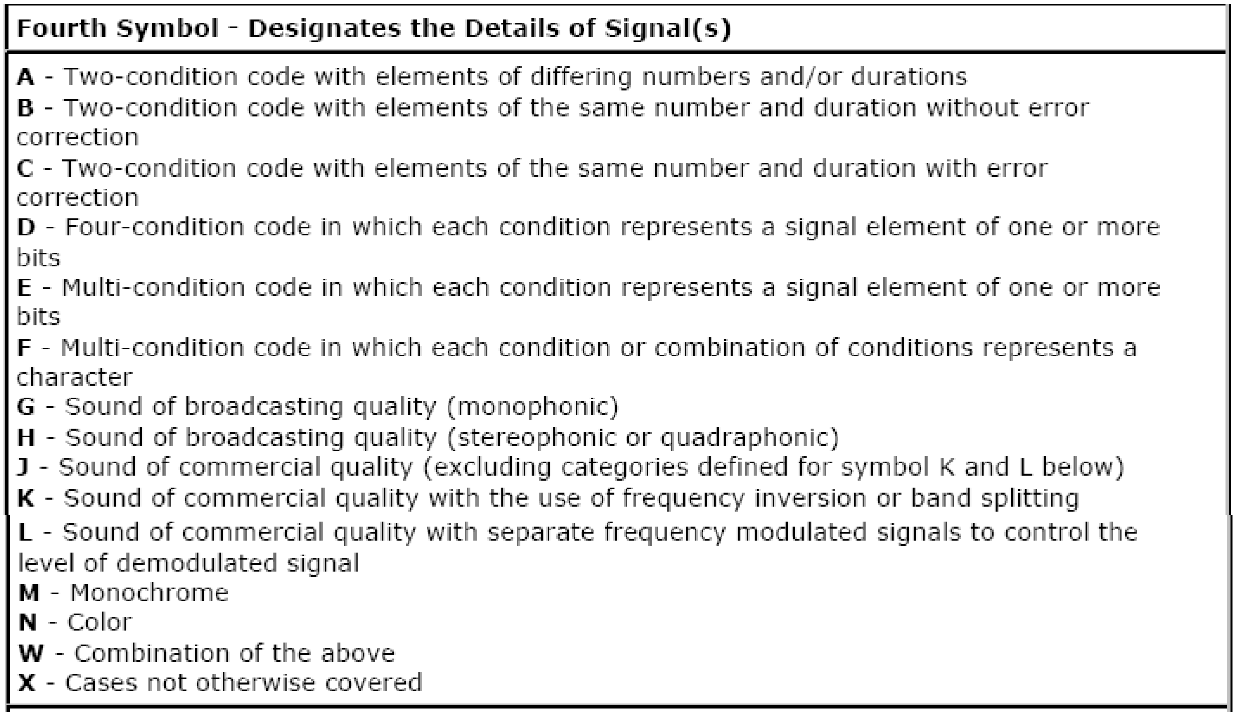
Combination
- **W** - Cases not covered above in which an emission consists of the main carrier being modulated, either simultaneously or in a preestablished sequence, in a combination of two or more of the following modes: amplitude, angle, pulse

Other
- **X** - Cases not otherwise covered

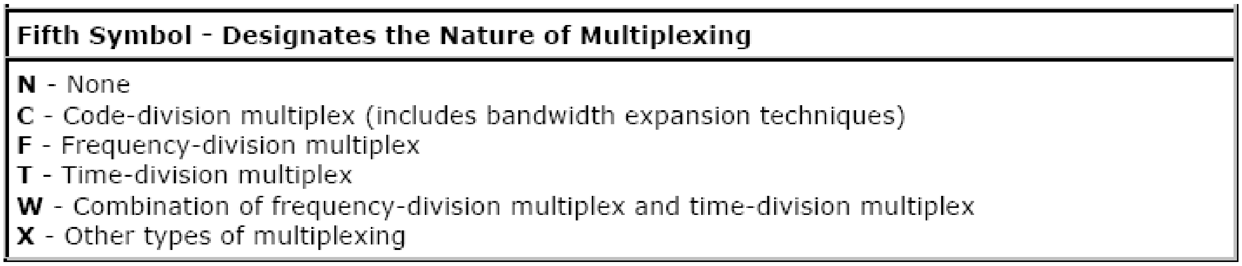


Second Symbol - Designates the Nature of Signal(s) Modulating the Main Carrier
0 - No modulating signal
1 - A single channel containing quantised or digital information, not using a modulating subcarrier. (Excludes time-division multiplex)
2 - A single channel containing quantised or digital information, using a modulating subcarrier
3 - A single channel containing analogue information
7 - Two or more channels containing quantised or digital information
8 - Two or more channels containing analogue information
9 - Composite system with one or more channels containing quantised or digital information, together with one or more channels containing analogue information
X - Cases not otherwise covered

Third Symbol - Type of Information to be Transmitted [a]
N - No information transmitted
A - Telegraphy - for aural reception
B - Telegraphy - for automatic reception
C - Facsimile
D - Data transmission, telemetry, telecommand
E - Telephony (including sound broadcasting)
F - Television (video)
W - Combination of the above
X - Cases not otherwise covered.[b]



Fourth Symbol - Designates the Details of Signal(s)
A - Two-condition code with elements of differing numbers and/or durations
B - Two-condition code with elements of the same number and duration without error correction
C - Two-condition code with elements of the same number and duration with error correction
D - Four-condition code in which each condition represents a signal element of one or more bits
E - Multi-condition code in which each condition represents a signal element of one or more bits
F - Multi-condition code in which each condition or combination of conditions represents a character
G - Sound of broadcasting quality (monophonic)
H - Sound of broadcasting quality (stereophonic or quadraphonic)
J - Sound of commercial quality (excluding categories defined for symbol K and L below)
K - Sound of commercial quality with the use of frequency inversion or band splitting
L - Sound of commercial quality with separate frequency modulated signals to control the level of demodulated signal
M - Monochrome
N - Color
W - Combination of the above
X - Cases not otherwise covered



Fifth Symbol - Designates the Nature of Multiplexing

N - None
C - Code-division multiplex (includes bandwidth expansion techniques)
F - Frequency-division multiplex
T - Time-division multiplex
W - Combination of frequency-division multiplex and time-division multiplex
X - Other types of multiplexing

Examples of these symbols taken from [MCEB_09] are shown in the following where the first message is a single-sideband suppressed carrier (J) carrying analog information (3) for telephony (E) with an occupied bandwidth of 3.00 kHz. The second message denotes a frequency modulated signal (F) with no modulating signal (0) which implies no information being transmitted (N) with no further details (A) and no multiplexing (N) with an occupied bandwidth of 3.50 MHz.

```
<EmsDesignator emsClass="J3E" bandwidth="3K00"/>
<EmsDesignator emsClass="PONAN" bandwidth="3M50"/>
```

Note that both integer and floating point values are also supported and the schema typically places bounds on acceptable values, which can then be used while validating a message.

3.5 Classification

By having the Common Element introduce the classification level, each data set and each element can have a different classification level, which allows SSRF was to maintain multiple levels of classification, with levels of UNCLASSIFIED, RESTRICTED (non-U.S. datasets), CONFIDENTIAL, SECRET and TOP SECRET. Each dataset is required to have an overall security classification and further compartmentalization is supported.

3.6 Time

The common element also introduces effective date and expiring time for elements, which allows SSRF to reflect the time validity of the data. The basic format for the DateTime in 20-24 characters as YYYY-MM-DDThh:mm:ss[.ddd]Z (year-month-day "T" hours: minutes: seconds. milliseconds"Z"), where the milliseconds part is optional. For SSRF to manage and synchronize systems around the world, all times are referenced to Zulu time (the ‘Z’). Also as a convention, the value 00:00:00Z is reserved to indicate that time is not an issue.

3.7 Handling Contextual Disagreements

It is interesting to note that SSRF anticipates the possibility of contextual incompatibilities [MCEB_09].

> As SSRF is meant to exchange information (datasets) between different data repositories, a dataset identifier may only apply to the data in the current message and in the data repository from which the data is extracted. For example, an assignment request may refer to a Location record which is known in the local (requester) data repository, therefore the

request contains a valid Location identifier with respect to this data repository, however not being known on the addressee side.

This message would be valid at the sending side but invalid at the receive side, since the record identified by this **serial** would not exist.

Two solutions are possible to this situation:

- By further handshaking: The addressee may send back a[n] Administrative message, requesting the missing datasets.
- By anticipation: The various software tools may offer an option to send either the dataset identifiers only of all the referred data, or to send a full copy of all referred records.

3.8 Radio Domain Information

SSRF provides an extensive language for describing radio transmit and receive properties. This includes emitted signal properties as shown in Figure 15, antenna properties, frequency allotments and assignments, interference reporting, Equivalent Isotropically Radiated Power (EIRP), tunable ranges of specific devices. Other information includes force movement patterns (for spectrum planning), operating regions, constraints imposed by the Host Nation. All-in-all, the vocabulary for SSRF extends over 600 pages in [MCEB_09].

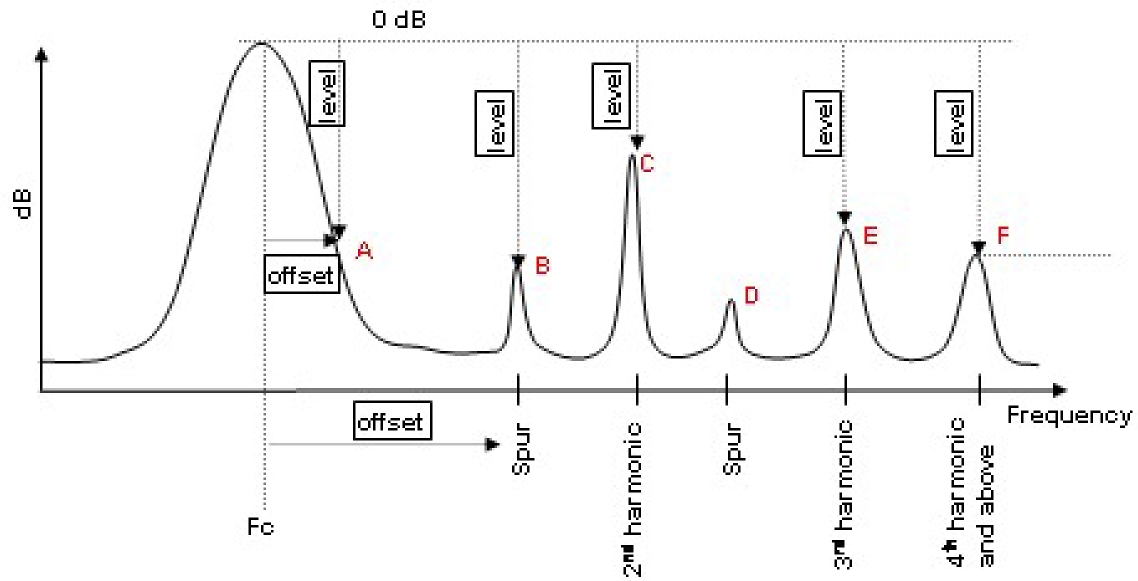


Figure 15: Waveform Descriptive Information in SSRF for transmit emissions. From Figure II-12.h in [MCEB_09].

3.9 Lessons Learned from SSRF for IPA

The following are key insights for information processing that can be gleaned from SSRF.

- **Inheritance of common elements across multiple levels allows for a more compact dictionary.**
  Despite the fact that [MCEB_09] is a 632 page document, the use of inheritance makes the definition of SSRF much more efficient. Enabling such a process requires identifying which attributes are common to all elements, such as time and classification with SSRF.

- **Relatively simple methods can be used to recognize incompatible context**
  SSRF uses the verification of messages against defined schema and known datasets to detect when a message's context is not understood by the recipient. Other checking is performed by listing schema in the message header. In SSRF, a context-matching failure can trigger a request from the recipient to the sender to transmit the unknown schema or dataset. More sophisticated methods could be developed where the syntax is satisfied but the message is used in a way that is not semantically understood.

- **The notion of a time validity period appears to be common to many pieces of contextual data**
  The data associated with every other element in SSRF can be assigned a start time and an end-time, much as was done with CoT, since effective and expiring dates are defined in the Common element.

- **Even systems with a standardized language will need to interface with systems using different languages**
  SSRF is able to create well-defined boundaries and translation processes as was illustrated in Figure 12, because SSRF recognizes that it will still have to interface with systems that do not implement SSRF, e.g., NTIA (OSMDD) and NATO (SMADEF-XML).

- **Multiple domains of information can be synthesized into a common language**
  SSRF combines radio domain information (e.g., signal properties) and tactical information (e.g., troop movement) into a single language. Such a combination of information from multiple domains will likely be critical to many emerging CR applications.

3.10 Standard Spectrum Resource Format References

[Anderson_07] J. Anderson, "The New Wizard War: Challenges and Opportunities for Electronic Warfare in the Information Age," November 6, 2007, Naval War College. Available online: http://dodreports.com/pdf/ada476476.pdf

[DoD_11] Department of Defense Instruction, "Electromagnetic Spectrum Data Sharing," **Number 8320.05**, August 18, 2011. Available online: http://www.dtic.mil/whs/directives/corres/pdf/832005p.pdf

[Lynch_09] B. Lynch, "Spectrum Encroachment and other challenges," JEPAC 2009, July 28, 2009. Available online: http://www.myaoc.org/EWEB/images/aoc_library/Events/2009/M&S/Briefs/14-Lynch%20JSC%20Slides.pdf

[MCEB_09] Military Communications-Electronics Board, "Standard Spectrum Resource Format (SSRF)," Version 1.2.4, MCEB Pub 8 02 Mar 2009. Available online:

4 1900.6

IEEE 1900.6-2011 is a standard for exchanging sensing related information approved April 22, 2011. The following is based on Draft 1 of IEEE 1900.6. While most TV White Space (TVWS) efforts have shifted to focus on geolocation database methods for enabling TVWS usage, 1900.6 still provides a good reference point on how cognitive radios can meaningfully share information in an actionable way.

4.1 System Objectives

1900.6 is intended to facilitate the real-time sharing of sensing information between disparate cognitive radio systems to enable sensing-based DSA. It is intended to allow collaborative sensing, sensor tasking, data archiving, and the ready integration of new CR technologies as they are developed.

Many initial efforts for deploying DSA in the TV bands relied on the use of spectrum sensing to detect the presence of primary users (protected spectrum incumbents). However, sensing by a single device is clearly inadequate due to hidden nodes, thus measurements needed to be taken by several different sensors. Further, increasing the number of unbiased independent observations of any phenomenon improves the receiver operating characteristic curve (the tradeoff of probability of detection versus probability of false alarm, e.g., errors of Type I (missed detection) versus Type II (erroneous detection)) of the detection / classification process. Thus early DSA research quickly identified that collaborative sensing would be critical to making sensing-based DSA practical. However, a standard would be needed to allow the many different devices, vendors, and Layer 1/2 standards being developed for the TV White Spaces to effectively share their sensing information and collaborate with one another. Filling this gap was the goal of 1900.6.

4.2 System Design

To realize this standard, 1900.6 defined[2]:

> "the interfaces and data structures required to exchange sensing-related information in order to increase interoperability between sensors and their clients developed by different manufacturers are defined in this standard. The logical interface and supporting data structures are defined abstractly without constraining the sensing technology, client design, or data link between sensor and client. [The standard further defines t]he entities involved and parameters exchanged in [sensor information exchange]. It further elaborates on the service access points, service primitives, as well as generic procedures used to realize this information exchange"

---

[2] From the abstract for 1900.6.

4.2.1 Architecture

To achieve these goals, 1900.6 builds on the system model shown in **Figure 16** with key terms defined in Table 2. 1900.6 is a server-client architecture that defines both the messages that can be exchanged and the expected behaviors upon receipt of messages. Interestingly, every element considered by 1900.6 – Sensor, Cognitive Engine (CE), and Data Archive (DA) (terms defined in Table 2) – is allowed to take the role of client. It is important to note that for 1900.6, "client" and "server" refer the direction of the flow of sensing related information (from a server to a client) and are not related to the use of service primitives and subsequent exchange of protocol messages required to realize the exchange of sensing related information. This also differs from traditional usage in that to fully implement a service, a single client may have multiple servers, e.g., for distributed sensing where a CE makes use of multiple Sensors (See Appendix D in [1900.6]). Further, all of these entities are “logical” entities in that they may be only a component of a larger device, may constitute a complete device, and multiple entities may be hosted on a single device, e.g., a CE with one or more Sensors on a single radio.

Table 2: Key 1900.6 Terminology. Excerpted from Section 3 in [1900.6]

1900.6 Term	1900.6 Meaning
Cognitive engine (CE)	“The portion of the cognitive radio system containing the policy based control mechanism and the cognitive control mechanism that has the knowledge about the current state and the set of attainable states of the reconfigurable radio platform”
Data archive (DA)	“A logical entity in which sensing related information obtained from spectrum sensors or other sources, as well as regulatory and policy information are processed and stored systematically. Note that the DA processing capability is limited to storing, retrieving, data format conversion and querying (fundamental data processing). Analyzing sensing related information for decision making is done by the CE.”
IEEE 1900.6 client	“An IEEE 1900.6 logical entity, application or device that receives sensing information and spectrum usage related information from an IEEE 1900.6 server. In general, the information exchange between IEEE 1900.6 client and service applies to sensing related information.”
IEEE 1900.6 server	“An IEEE 1900.6 logical entity, application or device that provides sensing information and spectrum usage related information to IEEE 1900.6 clients. In general, the information exchange between IEEE 1900.6 client and IEEE 1900.6 server applies to sensing related information.”
Sensor	“The portion of a radio system that performs sensing (see the definition of sensing in subclause 3.1) within a cognitive radio system. Sensors may also act as clients to other sensors.”
Sensing	“In the context of radio frequency spectrum, refers to the act of measuring information indicative of spectrum occupancy (information may include frequency ranges, signal power levels, bandwidth, location information, etc.). Sensing may include determining how the sensed spectrum is used”

The principle components of the 1900.6 architecture are shown in **Figure 16** and are formally defined in Table 2. The use cases for the three different interfaces shown in **Figure 16** are described in Table 3. A critical concept to 1900.6 interfaces is the 1900.6’s choice to not define the medium by which messages are passed; rather the content and meaning of the messages that will be passed. In this way, 1900.6 is intended to be agnostic to the DSA-enabled PHY / MAC standard it is supporting and can support multiple logical entities hosted on a single device.

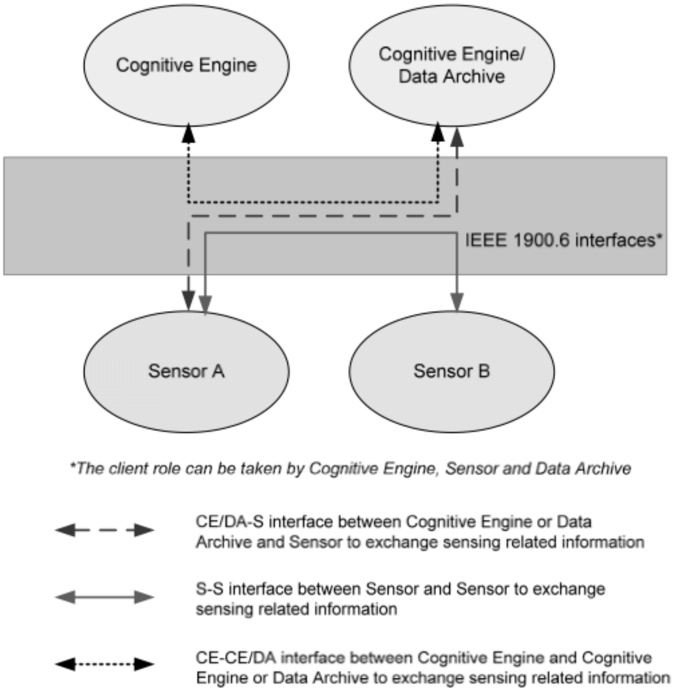


Figure 16: 1900.6 System Model. From Figure 1 in [1900.6][3]

Table 3: Use Cases for Interfaces in 1900.6. Excerpted from Section 1 in [1900.6]

Interface	Usage Scenario
CE/DA-Sensor	"exchanging sensing related information between a CE or DA and a Sensor. As an example, the CE/DA-S interface is used in scenarios where a given CE or DA obtains sensing related information from one or several Sensors or a given Sensor provides sensing related information to one or several CEs or DAs."
Sensor-Sensor	"when multiple Sensors exchange sensing related information for distributed sensing"
CE-CE/DA	• "where CEs exchange sensing related information for distributed sensing. " • "where a CE obtains sensing related information and or policy/regulatory information from a DA"

4.2.2 Service Access Points

Service Access Points (SAPs) are critical to achieving 1900.6's goal of being independent of the media over which its messages are exchanged. Traditionally, a SAP is a conceptual location where one Open Systems Interconnect (OSI) layer can request the services of another OSI layer.

---

[3] Clarifying the Sensor-Sensor interface, Appendix D of [1900.6] addresses distributed sensing and includes the following text. "*Distributed stand alone sensor type III is an example of using the IEEE 1900.6 logical interface to exchange sensing control and sensing information between a client spectrum sensor and another spectrum sensor. The client sensor is a sensor with application for either data fusion or relaying. It obtains sensing information from other sensors and merges it with its own sensing information and forwards it to a CE. The smart sensor could be an integrated sensor manufactured to provide the above functionality or a smart sensor capable device introduced in Figure D.4* [not included in this document]. *This particular implementation example of IEEE 1900.6 logical interface assists distributed spectrum sensing where sensors share their sensing information to make optimum local decisions before forwarding the final result to the CE. It also further assists relaying function of sensing information*."

By providing these black-box interfaces, a SAP abstracts away the implementation of an OSI layer so that when one OSI layer (client) is making use of the services (server) of another OSI layer, the client does not have to know how the server implements the requested service, only that it can implement the service.

Similarly, 1900.6 defines the following three SAPs which reside in a control plane for device management as shown in Figure 17:

- Measurement SAP (M-SAP) - to control and gain information from spectrum sensing
- Communication SAP (C-SAP) – to exchange sensing related information
- Application SAP (A-SAP) – for non 1900.6 entity use

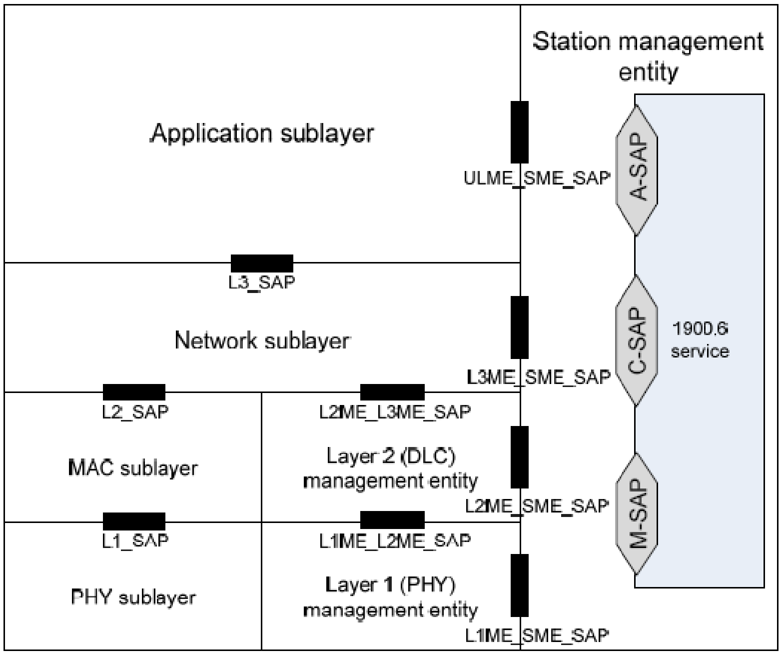


Figure 17: Example implementation of 1900.6 in station management entity in control plane. From Figure 9 in [1900.6]

1900.6 defines generic primitives and methods for these three SAPs to provide the following classes of services.

Table 4: SAPs and Service Classes in 1900.6[4]

1900.6 SAP	Purpose	Service Classes
M-SAP	"access IEEE 1900.6 compliant services provided by the station's hardware and/or firmware to control the spectrum measurement module (such as a collocated physical spectrum measurement module, i.e., ADC/DAC, filtering, signal conditioning, etc.), and to acquire spectrum measurement data. For example, a station (terminal) utilizes its RF interface during idle times for spectrum measurement and provides RF spectrum data to collocated IEEE 1900.6 "sensor" entities that are registered at the local M-SAP."	• Measurement capabilities discovery services • Measurement configuration discovery services • Measurement configuration services • Information services
C-SAP	"sensing related information (sensing information, sensor information, control information, and requirements derived from regulation) exchange between Sensors and their clients. The client role can be taken by a Sensor, a CE or a DA. It abstracts communication mechanisms for use by IEEE 1900.6 services through defining a set of generic primitives and mapping these primitives to transport protocols."	• Sensing related information send service • Sensing related information receive service • Information services
A-SAP	"services to an IEEE 1900.6 service user which is not an IEEE 1900.6 entity by itself… utilize sensing related information for its purpose, e.g., for policy investigation and analysis of spectrum usage. The A-SAP may provide functions to set-up a configuration of IEEE 1900.6 entities (e.g., Sensors and CE), to configure these for collaborative sensing, to start the data acquisition and processing (e.g., policy processing), and to obtain the results of IEEE 1900.6 processing in order to configure the RF interface accordingly."	• Sensor discovery service • Sensing related information access service • Management and configuration service • Information services

It is important to note that 1900.6 is providing mechanisms to both exercise sensing capabilities and to discover capabilities of 1900.6 compliant devices. Also notionally, the A-SAP provides an interface by which devices that do not implement the full 1900.6 service suite can still gain access to the services of a 1900.6 device.

4.2.3 1900.6 Services

For each SAP, 1900.6 defines a collection of services that the SAP will support. For each service, 1900.6 specifies:

- The function (what the service does)
- The semantics of the service primitive (interface)
- "When Used" (the situation when the service is expected to be used)
- "Effect of receipt" (what the user of the service experiences)

As far more service definitions are provided than can be covered here, we only reproduce one service definition to illustrate the kind of descriptions used in 1900.6's information exchange.

Note that in the example given below, datasheet elements are referred to. These are members of a much larger structure and only a subset of the information is exchanged in this message, presumably to reduce overhead when less information is required.

---

[4] It is believed by the author that all SAPs must be implemented for an entity to be 1900.6 compliant. But this is not entirely clear from the document. As such, an industry group may be needed to standardize profiles to ensure interoperability in practice and not just in theory. Such an exercise was similarly performed by the WiFi Alliance for 802.11.

5.3.1.2.2 Get_Sensor_PHY_Description.response

Function

This primitive returns the results of the request to obtain the description of the PHY profile of the spectrum measurement module.

Semantics of the service primitive

```
Get_Sensor_PHY_Description.response
                                      (
                                      SensorPHYProfileID,
                                      Status,
                                      Datasheet.CalibrationData,
                                      Datasheet.CalibrationMethod,
                                      Datasheet.ChannelFiltering,
                                      Datasheet.DynamicRange,
                                      Datasheet.NoiseFactor,
                                      Datasheet.PhaseNoise,
                                      LockStatus
                                      )
```

Parameters

Name	Type	Description
SensorPHYProfileID	Unsigned integer	SensorPHYProfileID uniquely defines the PHY profile of the spectrum measurement module.
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
Datasheet.CalibrationData	String	Data used to calibrate a spectrum measurement module.
Datasheet.CalibrationMethod	String	Method to calibration the spectrum measurement module.
Datasheet.ChannelFiltering	String	The filtering function of the spectrum measurement module.
Datasheet.DynamicRange	Float	Dynamic range of signal detection of spectrum measurement module.
Datasheet.NoiseFactor	Float	The ratio of the noise produced by the spectrum measurement model to the thermal noise.

Datasheet.PhaseNoise	Float	Phase noise produced by the spectrum measurement module.
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The entity is locked by a client for exclusive service. 0: The entity is not locked, has been unlocked by its client, or has been unlocked by a self-generated timeout event.

When used

The primitive is used in response to the Get_Sensor_PHY_Description.request primitive.

Effect of receipt

Upon receipt, the IEEE 1900.6 SAP user obtains the information related to the PHY profile of the spectrum measurement module.

4.2.4 Information Description

1900.6 further defines in much detail the structure of information exchanged between 1900.6 clients and sensors, the data types and structures used, the structure of control messages, and an object model for all parameters and commands exchanged between client and server.

1900.6 sensing information includes both information and meta-information. Types of information reported include position, time and confidence of acquisition, frequency band, energy, channel condition, time stamp of sensing, and local detection results. Meta-information includes the update rate of a sensing parameter and relative sensor positioning. The intent of meta-information is to ensure this information is known without the need to communicate such information with every sensing report.[5]

1900.6 control messages are organized into Command Groups and transmitted in the structure shown in Figure 18.

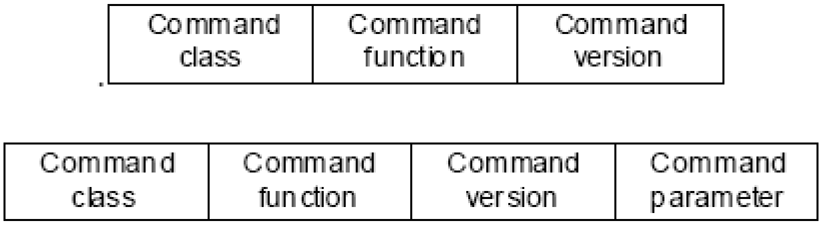


Figure 18: Command Group Structure. From Figure 10 in [1900.6]

---

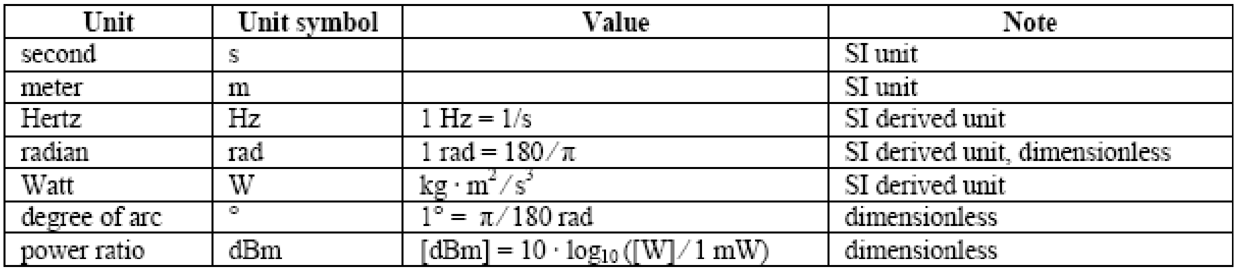
[5] This use of “meta-information” does not completely line up with the IPA’s use of the term “meta-information.”

These are further explained as follows in [1900.6] (emphasis added)

> “**Command groups** are either associated with a certain 1900.6 logical entity, are generic for all entities, or are manufacturer specific. The latter group shall use a specific command structure so that commands of this group are only recognized within the scope set by the device manufacturer. The **command class** specifies a distinct group to which a certain command belongs to. The **command function** unambiguously defines the purpose and scope of validity of a command as well as the subsequent command structure (i.e., if it requires additional command parameters or not). The **command version** indicates the version of the command and the command structure following the command class, function and version fields. This field is used to provide backward compatibility for further releases of the IEEE 1900.6 standard. Some control commands might be accompanied by **parameters** to narrow the required action. For example, starting a sensing activity might require setting start time, start frequency, scanning bandwidth, frequency increment, etc. to completely specify the action requested.”

To reduce implementation confusion, the units for exchanged information are standardized as shown in Table 5.

Table 5: Units used in 1900.6. From Table 1 in [1900.6]



Unit	Unit symbol	Value	Note
second	s		SI unit
meter	m		SI unit
Hertz	Hz	1 Hz = 1/s	SI derived unit
radian	rad	1 rad = 180 / π	SI derived unit, dimensionless
Watt	W	$kg \cdot m^2 / s^3$	SI derived unit
degree of arc	°	1° = π / 180 rad	dimensionless
power ratio	dBm	$[dBm] = 10 \cdot \log_{10}([W] / 1\ mW)$	dimensionless

All information is categorized into the classes of information shown in Table 6, from which Unified Modeling Language (UML) representations of these classes are defined, an example of which (for the Sensing Control Class) is shown in Figure 19.

Table 6: Classes of Information defined in [1900.6]

Class of Information	Content
Sensing	"Sensing parameters indicate the measurement output at the spectrum sensor and other associated parameters that augment the measurement data. Some parameters may appear as control parameter when issued by the client to configure the sensor and they may also be represented as sensing parameter[s] when they are issued as measurement output by [a] sensor. An example of such parameters may include … bandwidth and time stamp."
Sensing Control	"Sensing control information is used to optimize the spectrum sensing and the procedure of obtaining sensing information. Sensing control parameters are generated by the IEEE 1900.6 client. These parameters shall be used to realize the following two major functions. • Sensor configuration according to the demands of the application and to the measurement process. • Topological configuration of multiple sensors according to the demands of information exchange between sensors and between sensors and client."
Sensor	"Specifying standard sensor parameters would be helpful to identify the sensor capability and optimize the measurement request or configure the sensor. The sensor parameters may also be available in a local database on the client. But it can also be directly requested from the sensor. Depending on the application selected sensor profile parameters could be requested"
Regulation requirements	"Specifying sensing range, accuracy requirements, granularity, required measurement bandwidth, repeat frequency, detection sensitivity, sensing permission and synchronization. The values of these parameters may vary according to regulation depending on the region or country of deployment."

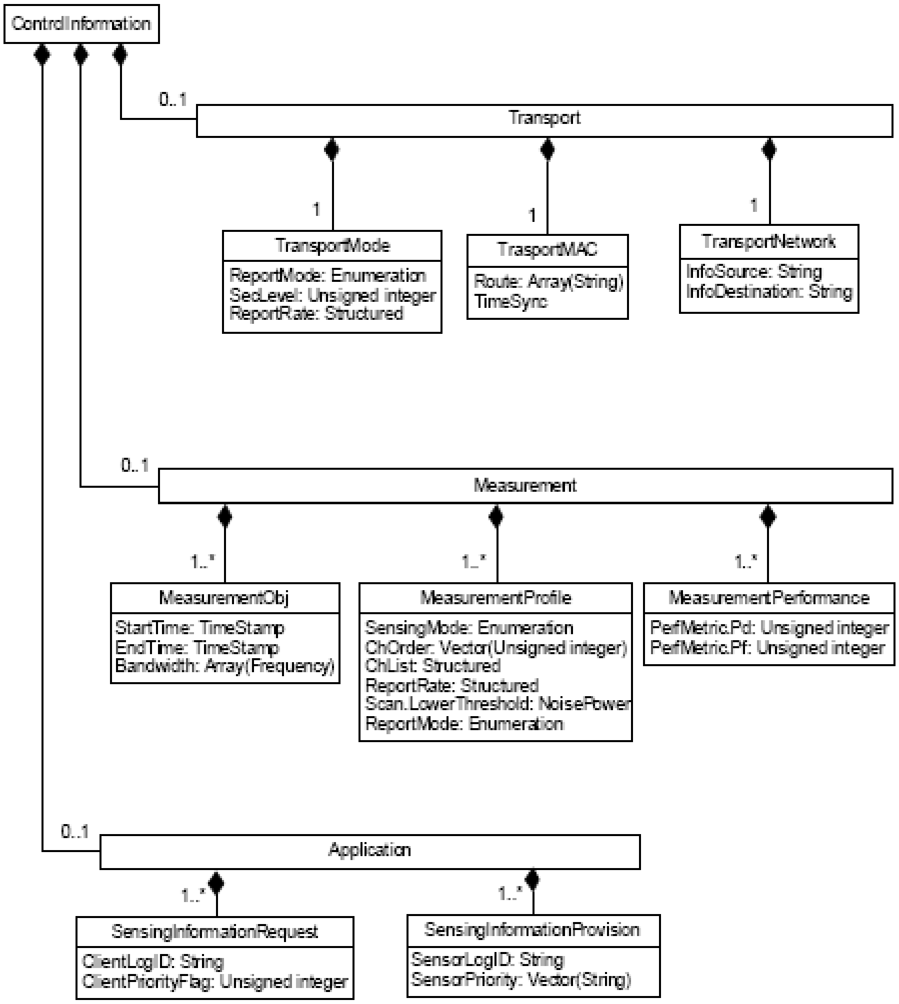


Figure 19: Sensing Control Class. From Figure 13 in [1900.6]

The format of parameters in 1900.6 messages are standardized specifically to support the sensing application. For instance:

- Table 7 lists the information that a client can use to request performance characteristics of a sensor.
- Table 8 provides the parameters that can be used to maintain 1900.6 system synchronization. Note that 1900.6 does not specify a time synchronization procedure; it merely provides mechanisms to support time synchronization.
- Table 9 lists the parameters a manufacturer can include in a datasheet for a sensor. This information could then guide the operation of a CE looking to task Sensors.

It is also useful to note that many of the parameterizations reference other information representation standards. For instance, geolocation information uses the WGS 84 reference coordinate system.

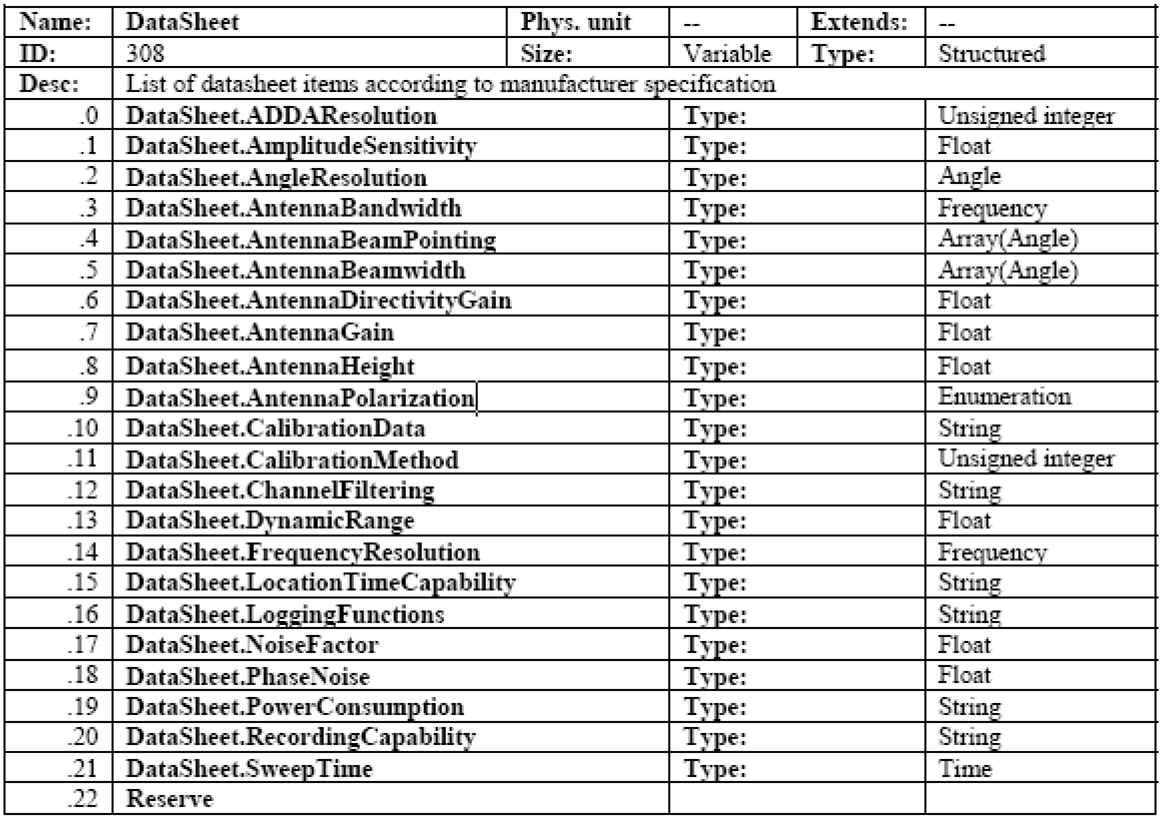
Table 7: PerfMetric Parameters

Name:	**PerfMetric**	**Phys. unit**	%	**Extends:**	--
ID:	106	**Size:**	Variable*	**Type:**	Array(Unsigned integer)
Desc:	Parameter that indicates the quality of sensing.				
.0	**PerfMetric.Pd**	**Type:**	Unsigned integer	Note 1	
.1	**PerfMetric.Pfa**	**Type:**	Unsigned integer	Note 2	
Note	1.	When the PerformanceMetric.pd is specified sensors perform sensing by setting the probability of detection according to this value. Probability of detection is bounded between 0 and 1			
	2.	When the PerformanceMetric.pfa is specified sensors perform sensing by setting the probability of false alarm according this value. Probability of false alarm is bounded between 0 and 1			
	*	In some cases, only one of the information elements could be sufficient to describe the performance.			

Table 8: TimeSync Parameters

Name:	**TimeSync**	**Phys. Unit**	--	**Extends:**	--
ID:	201	**Size:**	3	**Type:**	Structured
Desc:	Sync command for synchronization				
.0	**TimeSync. ClockAddress (ClientLogID)**	**Type:**	---		
.1	**TimeSync.MaxError**	**Type:**	integer	Note 1	
.2	**TimeSync.Rsecond**	**Type:**	Rsecond		
Note	1	Max error could be defined as tolerable error where it is given by the ratio of time offset divided by reference time			

Table 9: DataSheet Parameters



Name:	**DataSheet**	**Phys. unit**	--	**Extends:**	--
ID:	308	**Size:**	Variable	**Type:**	Structured
Desc:	List of datasheet items according to manufacturer specification				
.0	**DataSheet.ADDAResolution**		**Type:**		Unsigned integer
.1	**DataSheet.AmplitudeSensitivity**		**Type:**		Float
.2	**DataSheet.AngleResolution**		**Type:**		Angle
.3	**DataSheet.AntennaBandwidth**		**Type:**		Frequency
.4	**DataSheet.AntennaBeamPointing**		**Type:**		Array(Angle)
.5	**DataSheet.AntennaBeamwidth**		**Type:**		Array(Angle)
.6	**DataSheet.AntennaDirectivityGain**		**Type:**		Float
.7	**DataSheet.AntennaGain**		**Type:**		Float
.8	**DataSheet.AntennaHeight**		**Type:**		Float
.9	**DataSheet.AntennaPolarization**		**Type:**		Enumeration
.10	**DataSheet.CalibrationData**		**Type:**		String
.11	**DataSheet.CalibrationMethod**		**Type:**		Unsigned integer
.12	**DataSheet.ChannelFiltering**		**Type:**		String
.13	**DataSheet.DynamicRange**		**Type:**		Float
.14	**DataSheet.FrequencyResolution**		**Type:**		Frequency
.15	**DataSheet.LocationTimeCapability**		**Type:**		String
.16	**DataSheet.LoggingFunctions**		**Type:**		String
.17	**DataSheet.NoiseFactor**		**Type:**		Float
.18	**DataSheet.PhaseNoise**		**Type:**		Float
.19	**DataSheet.PowerConsumption**		**Type:**		String
.20	**DataSheet.RecordingCapability**		**Type:**		String
.21	**DataSheet.SweepTime**		**Type:**		Time
.22	**Reserve**				

4.3 State Diagrams

A key concept in 1900.6 is that it defines expected behavior in response to received messages in addition to the messaging itself. Towards this end 1900.6 defines the five states shown in Table 10 (though most systems will only implement 4 states, omitting the Simultaneous Communication and Data Gathering State). A 1900.6 entity moves between states when messages are received as illustrated in Figure 20.

Table 10: States defined in 1900.6

State	Operations
Initialization	“The initialization state includes functions that shall be executed before sensing information exchange, for example, obtaining control/application ID.”
Idle	“This is the state when there is no usage of the logical interface.”
Data Gathering	“The data gathering state includes functions that shall be executed for obtaining measurement results. It also includes functions that shall be executed for obtaining sensing information for control/application.”
Communication	“The communication state includes functions that shall be executed for transportation of sensing related information from an IEEE 1900.6 entity to another IEEE 1900.6 entity using the communication subsystem.”
Simultaneous Communication and Data Gathering	“In the simultaneous communication and data gathering state communication for sensing information exchange and data gathering take place at the same time. For example, the measurement module performs spectrum measurement at frequency f1 and the communication subsystem is transporting sensing information at another frequency f2. The sensor may also receive sensing control information during execution of the measurement. Note that this state may not be present in simple systems performing only one action at a time (communication or data gathering).”

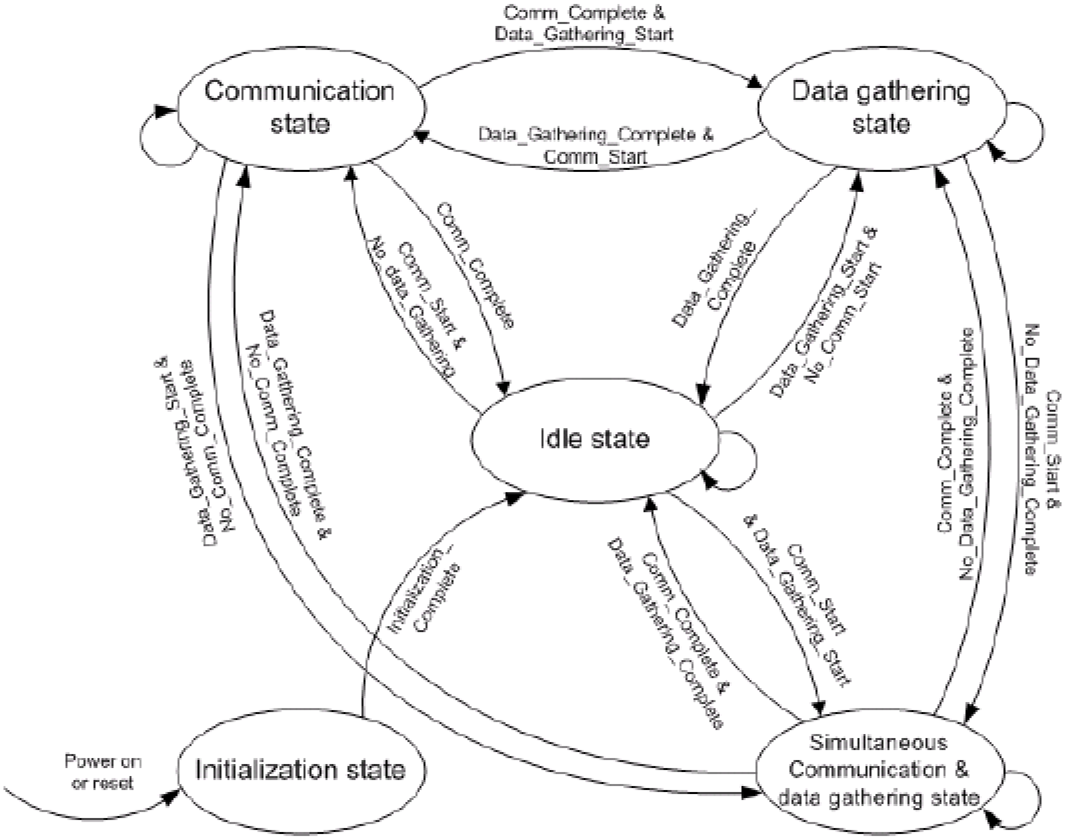


Figure 20: State diagram for sensing related information exchange. From Figure 17 in [1900.6]

4.4 Lessons Learned from 1900.6 for IPA

The following are key insights for information processing that can be gleaned from 1900.6.

- **Limiting scope and enforcing a common language facilitates communications across disparate groups**
  As with CoT, 1900.6 is only concerned with a very narrow scope; in this case the coordination of sensing operations and communication of sensing results. However, also like COT, extensions beyond the base (or common) level of communications are possible, e.g., in manufacturer datasheet fields and manufacturer-specific messages.

- **Efficiency, portability, and interoperability can be achieved with abstraction without anonymity.**
  The use of "logical entities" allowed 1900.6 to support multiple different device configurations (e.g., both CE and Sensor on the same device or across multiple devices) and to interoperate regardless of platform implementation. For a CE to make intelligent collaborative sensing decisions, however, it helps to know the provenance of the supplied information. Thus, mechanisms are included to allow applications to gain knowledge of device specifics.

- **Most information, including context, has a limited period of validity.**
  1900.6 notionally splits its communicated information into information and meta-information, or context, and then allows different messaging periods. For instance a datasheet might be transmitted only initially, but related information about a detection / classification event, such as imputed probability of detection or false alarm varies more and can be communicated more frequently.

- **Many of the application-specific protocols surveyed share common building blocks that are reused and extended for the application at-hand.**
  A general purpose intelligent agent wishing to communicate and process information will benefit from an ability to synthesize and extend protocols from common components (perhaps as a protocol factory). It may be possible to identify a core set of these components for use in an extensible information processing implementation with the synthesis informed by a meta-language that describes the use and dependencies of these components and inputs on what is required for the application. Examples of these common blocks include time and location information.

  In constructing (or adapting) these protocols, the relative terseness and expressiveness of the protocol can be adjusted for the network state and operating conditions (e.g., channel quality).

- **Defining a complete language, even for a narrowly focused application, is difficult. Mechanisms should be included to enable a language to evolve while maintaining backwards compatibility.**
  In reviewing the first draft of 1900.6, the CRWG found the draft to be simultaneously over-specified, possibly leading to confusion in implementations, and under-specified,

"depending on the application". This is not unusual as conveying information in an unambiguous yet efficient manner is a difficult task. Most languages (like software!) continue to evolve as implementing and using the language reveals issues and leads to ways in which the language could be improved. However, if we permit our CR information exchange protocols to evolve, whether via human or machine directed evolution, then care should be taken to maintain some degree of backwards compatibility.

4.5 IEEE 1900.6 References

[1900.6] IEEE, "IEEE P1900.6™/D1 Draft Standard for Spectrum Sensing Interfaces and Data Structures for Dynamic Spectrum Access and other Advanced Radio Communication Systems," April 2010.

5 Extensible Markup Language (XML)

XML, or e**X**tensible **M**arkup **L**anguage, is intended to provide a means for communicating textual material between disparate systems. Since its initial standardization in 1998, XML has grown to be a very popular tool due to its extensibility to many different applications and relative ease of machine implementation including Cursor on Target (Section 2).

5.1 XML Objectives

XML was developed rather quickly (the bulk of the work was completed in twenty weeks) to achieve the following objectives [XML_Tutorial]:

- Internet usability
- General purpose usability
- SGML compatibility[6]
- Facilitate the easy development of processing software
- Minimization of optional features
- Legibility (i.e., human readability)
- Formality
- Conciseness
- Ease of Authoring.

However, several of these objectives are naturally at odds. e.g., to be both usable in a general purpose sense and concise are conflicting objectives, as is convenience for machine processing and convenience for human reading. These conflicting objectives naturally led to a number of drawbacks, such as providing a seemingly endless set of alternate uses.

Ultimately, these objectives led to the XML standard being purely syntactical with semantics defined outside of the standard. This meant that any machine could easily process an XML text file, but the text could not be understood on its own, requiring additional context (schemas) that

---

[6] SGML, or Standard Generalized Markup Language, is an important precursor to XML.

vary from application to application. In effect, every new combination of the common XML syntax with a new schema defines a new markup language.[7]

5.2 XML Structure

As noted before, XML, the standard, defines only the syntax of a text file, not the meaning of the text. A very simple example of an XML fragment used just for marking up simple text is:

```
<Forumtag>
  <emp><ital>WINNF</ital></emp> is <emp>SDRF 2.0</emp>
</Forumtag>
```

Assuming an HTML-like schema, the expected printed result would be:
    WINNF is **SDRF 2.0**

In this case the process used to interpret the XML would have to know that "emp" means emphasis (bolding), and "ital" is the command for italics. That information is external to the XML specification and is typical of the support infrastructure normally found with XML. Again, since XML is semantically void, if provided a different schema, the content could have been processed entirely differently. Further, XML supports the entire Unicode character ser so the the fragment could have been expressed in any other language's character set and can even mix language sets.

Reviewing the fragment, there are two types of information being provided – *markup* and *content*. In this example, the content is "WINNF is SDRF 2.0", while the remaining characters are the *markup* that provide a set of instructions on how to process the content (in this case the relevant bolding and italics).

Markup information can be further classified as follows.

A **tag** is a basic construct that begins with the symbol "<" and ends with the symbol ">". Tags can be start-tags `<type>`, endtags `</type>`, or empty tags `<newpage/>`.

An **element** is a part of a document that begins with a start-tag and ends with the corresponding stop-tag. Elements can be nested to an arbitrary depth. Content may not contain either the characters "<" or "&", but those characters can be indicated by use of escape sequences "&lt" and "&amp" respectively.

An **attribute** is a name-value pair included in the start-tag where all values must be quoted and names can only appear once in an **element**. This implicitly illustrates another key feature of XML – it does not directly support data types as "2.0" could be a string or a float depending on the schema in use.

---

[7] With a bit of a stretch, XML can be seen as forming a Chomsky-like natural grammar on which languages are constructed.

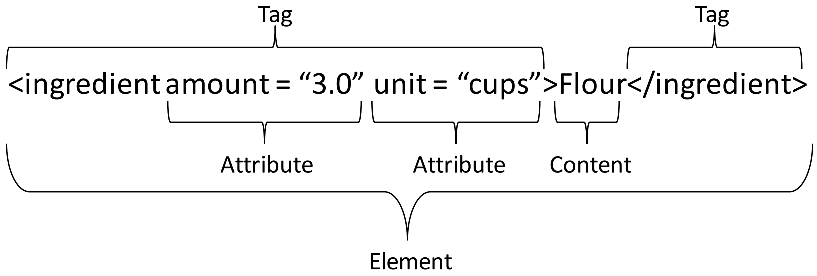


Figure 21: Illustrating the basic elements of an XML element. Modified from an example in [XML_tutorial]

To provided a common context for making the tags and content useful for processing, an **XML schema** places further constraints on the structure and content of selected XML files. Examples of popular XML schema include Document Type Definition (DTD), XML Schema (the capital *S* differentiates this specific instance from the more general concept of an XML schema), and RELAX NG. Various tools exist for validating that an XML file conforms to the rules of a schema.

Various tools are available for validating that an XML file conforms to the rules of a schema.

Depending on the application(s) supported, XML can also be used to define objects and convey rich fields of data. Considering again, Cursor on Target (which has its own schema and sub-schema), consider the following XML excerpt adapted from the Cursor on Target User Manual [Konstantopoulos_06], which provides examples of key features of XML with emphasis on data definition rather than display of text.

```
<?xml version='1.0' standalone='yes'?>

<event version="2.0"
     uid="J-01334"
     type="a-h-A-M-F-U-M"
     time="2005-04-05T11:43:38.07Z"
     start="2005-04-05T11:43:38.07Z"
     stale="2005-04-05T11:45:38.07Z" >

     <detail>
          <remarks>
               Provides a place to annotate CoT with free
               text information.
          </remarks>
     </detail>

<point lat="30.0090027" lon="-85.9578735" ce="45.3"
     hae="-42.6" le="99.5" />

</event>
```

This XML fragment can be explained as follows.

- The first line is a declaration of XML details.

- The tag labeled "`event`" has five "attributes" after the tag-name and before the closing character ">". These attributes set internal data values at the receiving end of the CoT message and together with the "`point`" element, these attributes carry the payload of the message. The CoT version of the XML processor at the receiving end would pass this information to the designated shooter or processor.

- The element <detail></detail> is null by default. However, this element can be extended by adding nested elements. For instance, provision for a remarks element to be inserted is in the CoT manual, as we have done here as an example. Thus via the *detail* field the CoT schema is itself extensible which leads to its broad applicability and potential compatibility issues.

5.3 Lessons Learned from XML for IPA

- **With data compression tools, redundancy and readability can be built into a language without a significant penalty for communications efficiency.**
  XML is frequently criticized for its verbosity. But in practice, this has been only a secondary consideration as where verbosity matters (low bandwidth networks), compression can reduce redundancy prior to transmission and restore it at the receiver.

- **Separation of context from payload improves transmission efficiency**
  In effect, XML provides a partial context for its payload, with significant amount of additional context being provided from other parts of the architecture (e.g., schema). In this way context only needs to be conveyed once, while payload can be inserted in each message.

  Additional improvements could be made by further separation of goals. For instance, the machine-readable version and the human readable version could be two presentations of the same source data (one is the alternate presentation of the other).

- **XML is a common base language ("tool") for other languages**
  From the perspective of human readability, this increase in scope results in many XML files which are either incomplete or overly complex. There are also many details in place to provide backwards compatibility with previous markup systems (e.g., HTML), which result in apparent inconsistencies.

XML has mixed goals of serving both as a data-description vehicle and procedure for document markup, and may represent the worst of both worlds. Future work in IPA would be well advised to consider separating those goals, and divorcing the machine-readable version from the user interface to provide better structure for the former and ease of use for the latter.

5.4 XML References

[Hunter_07] D. Hunter, et al., Beginning XML, 4th Ed. WROX Publishing 2007.

[St. Laurent_05] XML Pocket Reference, St. Laurent and Fitzgerald, O'Reilly, 2005

[XML_Tutorial] http://www.scribd.com/doc/49261218/XML-Tutorial

[W3C] http://www.w3.org/TR/xml/

6 Modeling Language for Mobility (MLM)

If wireless devices are to communicate and interoperate autonomously, they first need to speak a common language. This was the main message of a paper [Fette_08] published in 2008. This paper described a need for a standardized language (MLM, or Modeling Language for Mobility) with formal syntax and computer-processable semantics in which radios could express various aspects of communications, like their hardware and software capabilities (frequency bands, modulations, MAC protocols, access authorizations, etiquettes, configurations), networks available to a user (parameters, restrictions, costs), security / privacy issues (constraints, policies), information types (QoS, priorities), local spectrum (spectrum activity, availability, propagation properties), manufacturer's concerns (hardware and software licensing policy, versions, compatibility), types of users (authority, priority), and other.

6.1 Goals

The main reason for developing MLM is to provide the flexibility necessary for future generation radios. One way to achieve this goal would be to develop communication protocols that would be capable of exchanging control messages related to many more aspects than the current protocols can provide. This would lead to an increase in the size of the headers of the PHY layer packets, but it would still be limited by the size of the header fields. Another way would be to define a large vocabulary of control messages and then include such messages into the payload. Such control messages could be expressed in XML and each message would need to be interpreted by the radio software. This would give a great flexibility, but it would also require that radios had procedural code for interpreting each kind of control message. Yet another way is to give radios a formal language with computer-processable semantics in which any control message could be encoded, provided that it can be expressed in terms of an ontology shared by the radios. This approach does not require the existence of a separate procedure of each type of control message, but instead, a generic interpreter (an *inference engine*, or *reasoner*). This is the approach advocated in this paper.

The main advantages of this approach are: (1) a great flexibility in terms of the number of possible message and query types (practically unlimited), (2) an increase in communications efficiency due to the ability of sending only parts of messages, while the rest of information can be inferred locally by the reasoner based on the generic knowledge encoded in the shared ontology, (3) the ease of the adaptability to changes in the message/policy vocabulary (only the ontology needs to be modified, while the procedural code remains unchanged).

It is expected that the MLM language will provide new opportunities for various stakeholders. In particular, it is expected that by using MLM vendors of the radio software will be able to develop next generation interoperable radios independently.

In a later paper [Moskal_11] the progress made since the publication of [Fette_08] was reported. The main achievements during this period were: (1) Development of the first version of the Cognitive Radio Ontology (CRO); (2) Implementation of the Link Optimization demonstration in which two radios collaborate on establishing communication parameters of transmissions. The most important aspect of this demonstration is that the radios do not use any special signaling protocol, but instead exchange messages (in the payload) expressed in terms of the ontology. A reasoner on each radio infers then how to act, depending on the information (or request) received from the other node and on its own configuration (self-awareness).

6.2 CRO: Cognitive Radio Ontology

As mentioned earlier, the main objective of the MLM approach was to use a formal language, i.e., a language with a formal syntax and a formal, computer-processable semantics so that logical inference can be performed automatically on the language expressions by (generic) *inference engines*. Logical inference is possible only within a *formal system*, i.e., a system that includes a *formal language*, a *theory* (or axioms) and *inference rules*. Formal syntax means rules for determining whether a given expression is in the language or not (sometimes referred to as *legal sentences* or *well-formed formulas*). Formal semantics refers to *interpretations*, which are mappings from the language to a mathematical domain (a set of individuals) and from sentences to *truth values*. Theories are then represented by *axioms* – sets of sentences in the language. Inference rules are rules that can be applied to the axioms of a theory to derive new sentences, which then become part of the theory. A formal system should be *sound*, i.e., given a consistent set of true sentences, it derives only true sentences, i.e. sentences that map to the value “true” by the interpretation function. Another desirable, but unachievable, feature of a formal system is *completeness*, i.e., the ability to infer all possible true sentences using the rules of inference.

An *inference engine* can then take a set of sentences in a formal language and apply the inference rules of the formal system to derive new sentences. The most important aspect of this process is that the inference engine is generic, i.e., it can be applied to any set of axioms expressed in the given language. Thus the *queries* sent to the inference engine can be anything expressible in the formal language, rather than a predefined set. Thus the limit of inference is bound by the language, and not by a pre-defined set of functions and queries.

In the current status of MLM, all of the statements about the radio domain are expressed in the Web Ontology Language (OWL) [W3C_09] supplemented by rules (see discussion below). OWL is a formal language with model theoretic semantics. A number of generic inference engines for this language exist. In the MLM work the BaseVISor inference engine [Matheus_06] was used. This engine is freely available for research purposes, as are some others. BaseVISor is implemented in Java. It supports the OWL 2 RL dialect of OWL. OWL 2 RL includes most of the constructs of OWL 2, but additionally, it also supports the expression of user defined *policies* (collections of *rules*). The importance of rules stems from the fact that rules allow to express some more complicated relationships than just pure OWL can. BaseVISor is an inference engine applicable to OWL axioms and user-defined policies represented as rules. It is a forwardchaining rule engine since the rules are executed in the "forward" direction. That is, rules are applied for as long as there is new information that can be derived by rule applications. Since at the low level all the axioms are represented as triples, BaseVISor has been optimized for processing RDF- and OWL-expressed information.

OWL includes a number of generic concepts, like Class, Property, Individual and many more, that are useful to represent knowledge about various domains. To specialize OWL to a particular domain, one needs to develop an *ontology*. An ontology provides a shared vocabulary, which then can be used to represent more specific knowledge about a specific domain. An ontology captures the type of objects/classes (or concepts) that exist, their properties (attributes) and relations among concepts. Ontologies can be extended in two ways – by adding new classes and properties, or simply by annotating specific domain entities in terms of the classes and the properties in the current ontology. Note, however, that even if new classes and properties are added, the same generic inference engines can be used for drawing conclusions based upon information represented in the extended ontology.

The MLM Working Group (MLM WG) of the Wireless Innovation Forum developed and published a Cognitive Radio Ontology (CRO). CRO covers:

- the basic terms of wireless communications from the PHY and MAC layers;
- the concepts needed to express the use cases developed earlier by the MLM WG;
- partial expressions of the FM3TR waveform (structure and Finite State Machines) and the Transceivers Facilities APIs.

The CRO is formalized in the Web Ontology Language (OWL). The CRO includes 230 classes, 188 properties and various constraints. The rationale behind some design decisions of this ontology is partially described in [Li_09].

At the highest level, CRO classifies concepts as either *objects* or *processes*. Both objects and processes can have associated *quantities* (parameters) expressed as *values* in a given *unit of measure*.

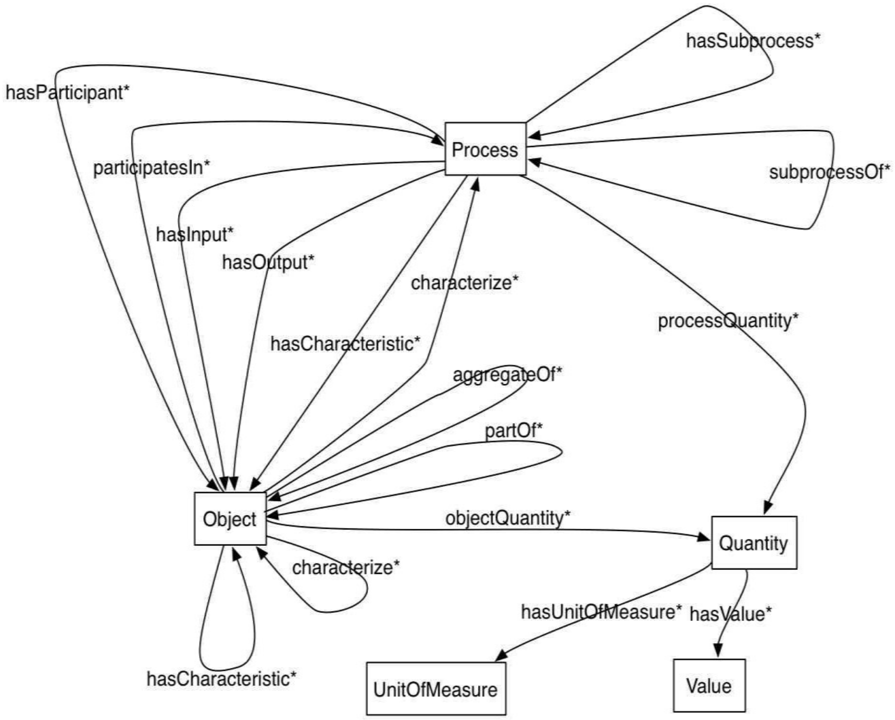


Figure 22: Top Level Portion of CRO. [Li_09]

These high level concepts are extended for applicability specifically for the cognitive radio domain, a small portion of which is shown in Figure 23. Concepts are related via various relations; here only a few are shown. For instance, *Decibel_Radio_B* is the *signalToNoiseRatio* of *SignalDetector_Radio_B*, which *hasValue mSNR_Ratio_B*. The most basic relation that links objects and processes is *participatesIn*, i.e., an object can participate in a process. In the example shown in Figure 23, objects *Radio_B* and *SignalDetector_Radio_B* both *participateIn* a process called *Transmitting_Radio_B*. Another basic relation is the relation of aggregation. For objects it is the *partOf* (or *aggregateOf*). For processes, it is *subProcessOf* (or *hasSubprocess*). In Figure 23, *Radio_B* is an aggregation of *SignalDector_Radio_B*. Note that *hasSubComponent* is a sub-property of *aggregateOf*; its inverse property is *isSubComponentOf.*

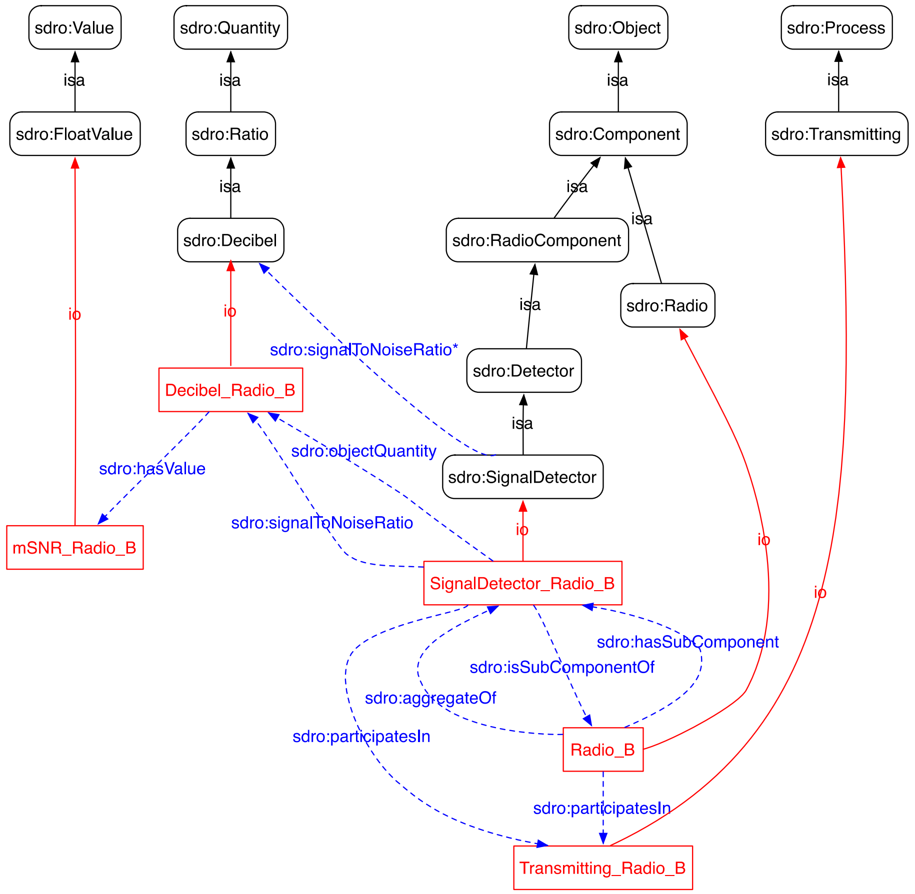


Figure 23: A small part of the CRO ontology

Qualities are the basic attributes or properties that can be perceived or measured. Qualities cannot exist on their own; they must be associated with either an object or a process. All the qualities in CRO have values and some qualities have units. The qualities without units are represented as data-type properties; the qualities with units are associated with a type of quantity.

Quantity is a representation of a property of an object. In other words, quantity is a representation of quality. For instance, a physical quantity represents a property of a physical object. Quantity carries three types of information:

- the type of the quantity (e.g., mass, length)
- the magnitude of the property (typically a real or integer number)
- the unit of measurement associated with the given magnitude (e.g., [*kg*], [*m*]).

In this ontology, quantity is a top-level class; it is further divided (sub-classified) into different types, such as length, frequency, time, etc. Each *quantity* is associated with a unit and a value. Note that there is no explicit *Quality* class in CRO. Instead, CRO uses *objectQuantity* and *processQuantity* to represent the quality of an object or a process, as shown in Figure 23.

6.3 A Case Study: MLM Based Link Optimization

To further illustrate the use of CRO in cognitive radio applications, consider the following case study where researchers working with CRO developed a link optimization use case, which was implemented on the GNU Radio USRP1 platform and exhibited at the 2010 Software Defined Radio Forum Technical Conference in Washington, D.C. [Li_10].

The general goal for this link optimization use case was to maximize the power efficiency (i.e., the information bit rate per transmitted watt of power), subject to a set of constraints. This is attained by fine-tuning the parameters in the transmitter and the receiver. Here, MLM provides a means to exchange the control messages between the transmitter and the receiver.
In short, the goal is to minimize the following objective function:

$$objFunc = 10^{\frac{PowdB}{10}} \times \left[\frac{528\left(1+\frac{m}{2^m - m - 1}\right)}{v} + trainPeriod \right]$$

In this objective function, there are four tunable variables (*knobs*):

- *PowdB* - the transmitter power
- *m* - the hamming code index
- *v* - the QAM modulation index
- *trainPeriod* - the length of the training sequence.

In this demonstration, each radio has an inference engine (System Strategy Reasoner, or SSR for short), as shown in Figure 24. The SSR has three types of inputs:

- the T-Box, which is the CR ontology that defines the common static knowledge shared by the two radios;
- the R-Box, which is the policies specified in MLM, describing how to react to particular situations;
- the A-Box, which are the dynamic facts about some knobs and meters that are only available as the radio is operating.

The two radios interoperate by exchanging control messages expressed in terms of CRO.

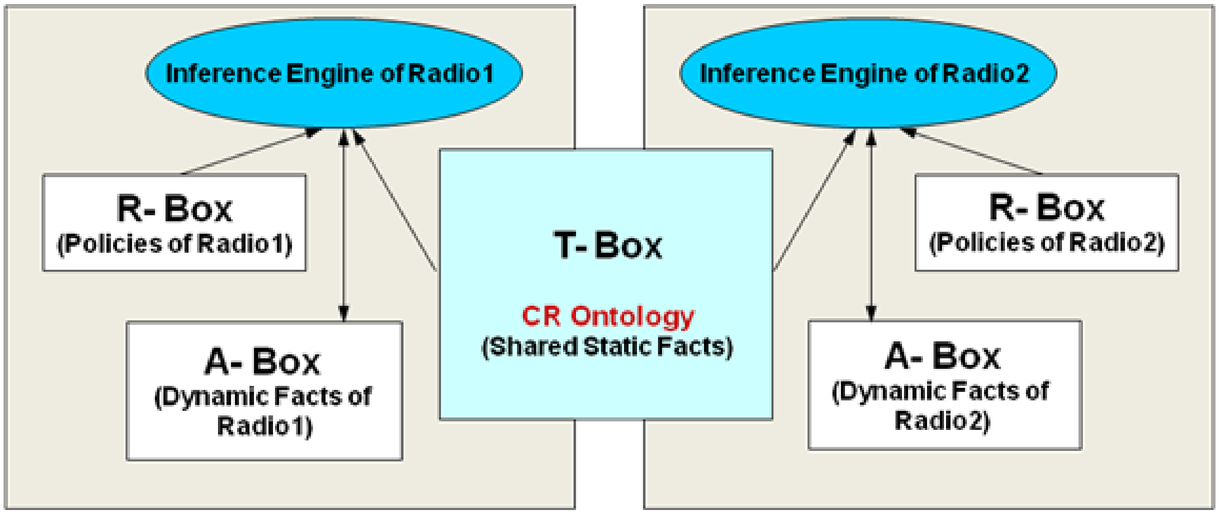


Figure 24: Inference Engine, Ontology and Policy

Figure 25 shows the architecture of this cognitive radio. All the incoming messages from the RF are first processed by the Radio Platform. Data messages are passed to the radio application (*Data Sink*), whereas control messages are passed to the SSR. Similarly, all the outgoing control messages are generated by the SSR and then passed to a buffer. Data and control messages are then merged and passed to the Radio Platform. After being processed in the Radio Platform, the messages are sent out through the RF.

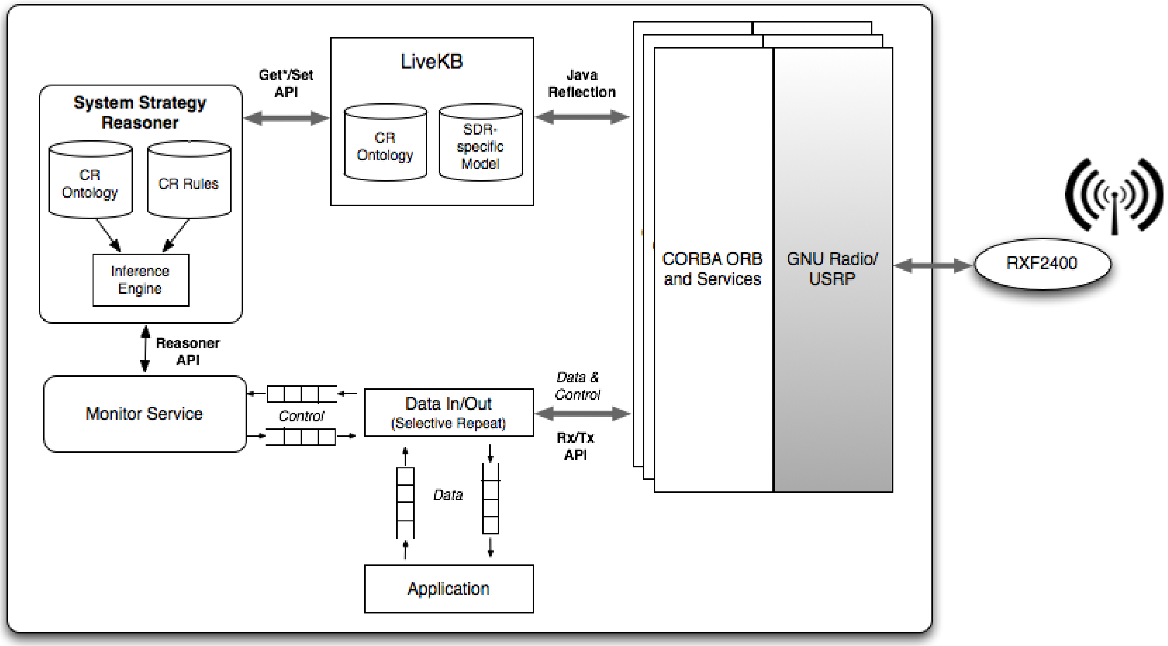
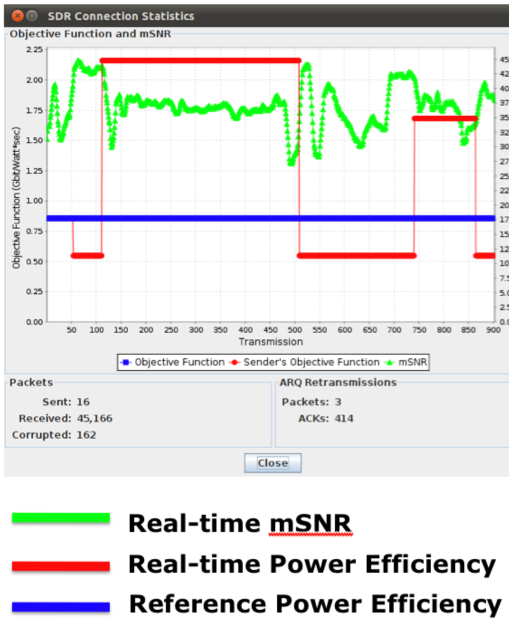


Figure 25: Architecture of Ontology-Based Radio

In addition to describing various parameters related to radio communications in terms of CRO, radios need to also implement *communication acts*. For instance, a node needs to understand whether a specific piece of information is a query, a request to perform a specific act (like set the value of a variable to a given value), or information about a variable's value of the transmitter. To achieve this level of interaction, the demonstration used the FIPA (The Foundation for Intelligent Physical Agents) Agent Communication Language (ACL) message structure, which provided the envelope for radio control messages. FIPA ACL is a specification that helps ensure interoperability between agents by providing a standard set of ACL message structures. The ACL part of the message indicates what kind of communication act it is. The inner part is the content of the control message described using MLM. The incoming control messages are first processed by the Monitor Service (MS), which unwraps messages, generates acknowledgments and other interactions with the MS of the other radio, and passes the MLM content to the SSR. The inference engine of the SSR interprets the content and makes decisions accordingly, passing replies to the MS for sending to the other nodes.

Figure 26 shows the demonstration results of the adaptation of the communications parameters of the two GNU radios. These plots show the mean SNR at the receiver and the power efficiency of the communications link. When the SNR falls out of the bounds of the predefined values, according to the policy of the transmitter, the transmitter power is adjusted in order to increase the overall power efficiency while keeping the mean SNR within the acceptable range. It can be seen that when the mean SNR at the receiver is too high, the two radios will exchange their parameters and a lower transmitter power is used at the transmitter, thus increasing the power efficiency. Conversely, when the mean SNR is too low, they will increase the transmitter power and thus decrease the power efficiency.



Packets
Sent: 16
Received: 45,166
Corrupted: 162

ARQ Retransmissions
Packets: 3
ACKs: 414

Figure 26: Implementation Results

6.4 Lessons Learned

- **A language can represent more than just What, Where, When**
  By using appropriate ontologies, MLM can be used to represent various aspects of the domain, not just the What, Where, When questions, as compared to CoT. In particular,

one may add aspects related to the Why and How questions. These last two would be important when one intelligent agent wishes another to be able to build upon or replicate its decision later. I.e., the recipient agent should understand not only that a decision was made, but why the decision was made so that when faced with similar (though perhaps different) problems, the reasoning process could be accelerated.

Note that MLM or any language considered for IPA applications may not address all possible questions needed for all possible communications. Further it may be that no machine-interpretable language exists that is capable of asking and answering all possible natural language questions and that for efficiency-sake, many questions may need to be out of scope for a particular IPA application.[8]

- **Languages can express policies**
  Policies are collections of rules. Thus policies on how to process information at the receiver can be expressed and then executed automatically by an inference engine. However for proper receiver operation, the conditions under which to execute these policies should also be met. MLM handles this issue by introducing rules as part of the policy that specify pre-conditions and post-conditions for when a policy applies.

- **A language can make use of both explicit and implicit context**
  While an ontology provides context for the information exchanged by the communicating nodes, MLM can also be used to represent context explicitly. Context would then be a collection of MLM expressions that would be added to the existing ontology or ontologies and then used by the inference engine. The efficiency of representing context, as compared to pure XML, is due to the fact that only some of the facts need to be expressed explicitly and transferred between the communicating nodes while the rest of the facts would be automatically inferred by the inference node at the receiver building on the context stored in the T-Box.

- **Languages can be modified via changes to the ontology and changes to procedures**
  By making procedures an explicit part of MLM, different behaviors can be defined by changing out the procedures. Likewise operation over different domains can be accomplished by changing out the ontologies. Policies can also be attached in the form of procedures (so called procedural attachments). Inference engines then can invoke the procedures according to the policies.

  For instance, CoT was defined for a specific domain with specific behaviors, necessitating that both be changed simultaneously. But in a language like MLM either the domain or behavior could be changed separately, though not without great care.

- **Development time is reduced by leveraging existing tools**
  Since MLM uses OWL and rules, any existing OWL development tools can be used for development of ontologies and policies.

---

[8] Recall the emphasis given to Scope in volume 1 to defining boundaries to IPA problems.

- **Development time can be reduced due to the use of generic inference engines**
  Similarly as XML, MLM is a declarative language. However, compared to XML, MLM provides the capability of logical inference using generic inference engines. XML, on the other hand requires the development of procedural code to process information annotated with specific XML tags.

- **By extending a language with procedures and processing capabilities at the sender and receiver (e.g., inference engines), both the original and enhanced capabilities can be simultaneously supported.**
  For instance, MLM is encoded using XML. Thus it inherits the various aspects, including advantages and disadvantages of XML, as discussed in the previous sections and could be interpreted by any XML parser enabled with the appropriate schema. But a richer capability set is enabled via the extensions that MLM adds on top of XML.

6.5 MLM References

[Fette_08] B. Fette, M. M. Kokar, and M. Cummings, "Next-Generation Design Issues in Communications," *Portable Design Magazine*, No. 3:20-24, 2008.

[Li_09] S. Li, M. M. Kokar, D. Brady, "Developing an Ontology for the Cognitive Radio: Issues and Decisions", *SDR Forum Technical Conference SDR '09*, Dec. 2009.

[Li_10] S. Li, M. M. Kokar, D. Brady, and J. Moskal, "Collaborative Adaptation of Cognitive Radio Parameters Using Ontology and Policy Approach," In *Software Defined Radio Technical Conference, SDR'10*. SDRF, 2010.

[Li_11] S. Li, J. Moskal, M. M. Kokar, and D. Brady, "An Implementation of Collaborative Adaptation of Cognitive Radio Parameters Using an Ontology and Policy Based Approach," *Analog Integrated Circuits and Signal Processing*, 69(2-3):283-296, 2011.

[Matheus_06] Matheus, C., Baclawski, K., & Kokar, M. (2006). BaseVISor: A Triples-Based Inference Engine Outfitted to Process RuleML and R-Entailment Rules. *The 2nd International Conference on Rules and Rule Languages for the Semantic Web.* Athens, GA: Proceedings.

[W3C_09] W3C. OWL 2 Web Ontology Language Direct Semantics. W3C Recommendation, 27 October, 2009. Available at: http://www.w3.org/TR/2009/REC-owl2-direct-semantics-20091027/.

[WinnF_10] WinnF Forum MLM Working Group, "Description of Cognitive Radio Ontology v.1.0", 2010. Available at http://groups.winnforum.org/d/do/3370.

7 Survey Insights and Future Work

While the surveyed technologies addressed a variety of different applications, there were nonetheless several common approaches employed and lessons that could be synthesized from this effort.

7.1 Common Threads from Surveyed Protocols

Most information, including contextual information, has a limited period of validity and scope.

Whether sensing information or the validity of a target in CoT, surveyed actionable information exhibited a time dependence. Thus we expect that an IPA system will need to model the notion of time. This can be broadened into a limiting scope for the information, whether geographic, e.g., for policies for SSRF or target locations for CoT, or for which entities the information applies, e.g., for policy languages.

This limited scope then allows disparate systems to communicate on common topics

Each of the surveyed standards limited how many topics could be communicated between the systems. 1900.6 is only concerned with the coordination of sensing operations and communication of sensing results. More sophisticated communications could be extended both in 1900.6, CoT, MLM, but the base (or common) level of communications are preserved.

Efficient coding can be derived from a limited dictionary (terse schemas)

Shorter words (generally one letter) can be used to convey meaning because of a limited dictionary. Further, the dictionary is segmented so that for each field there is only a limited number of words that can be used. This greatly simplified the implementation of machine-to-machine communications, which is at the heart of CoT, and facilitates the integration into different nations' systems.

Implicit context (conventions) related to time reduce bandwidth requirements

Through the use of conventions related to timing by which updates are only sent when information changes or when information becomes stale, many fewer messages can be sent. Loosely, this is akin to the savings achievable with event-based communications or computation.

Responsibilities for different types of information are segmented and fulfilled by nodes in different roles

In CoT, synthesized information (e.g., all sensor info within a specified area) is supplied by a server, but individual nodes are responsible for keeping the network apprised of their unique information via the server. Similarly, in SSRF different spectrum managers are responsible for managing different data sets.

7.2 Recommendations

It will be valuable to be able to recognize when a message is received in an incompatible context

All of the surveyed systems operated in highly constrained contexts. But as cognitive radios expand in scope and draw in information from more sources, maintaining such a tightly scoped context will be difficult. Thus it seems likely that CRs will occasionally receive out-of-context information. If a CR could be developed that recognizes when context mismatches occur (e.g., by tracing subsequent logic failures or by noting when specified dictionaries or schemas are not available as in SSRF), then errors could be trapped, though some errors will necessarily be uncatchable and the time required to detect logic failures may prevent correction in real time.

Further, this recognition could spawn requests for the correct or missing context, a process which we have loosely called context ARQ.

Defining a complete language, even for a narrowly focused application, is difficult. It is useful to include mechanisms to enable a language to evolve when maintaining backwards compatibility is an objective.

In reviewing the first draft of 1900.6, the CRWG found the draft to be simultaneously over-specified – possibly leading to confusion in implementations – and under-specified - "depending on the application". This is not unusual as conveying information in an unambiguous yet efficient manner is a difficult task. Most languages (like software!) continue to evolve as implementing and using the language reveals issues and leads to ways in which the language could be improved. But, if we permit our CR information exchange protocols to evolve, whether via human or machine directed evolution, then the impact on backwards compatibility should be considered.

A multi-layered multi-dimensional approach to context structure can enhance information system architecture.

- Layering limits the scope of processes and breaks larger communications problems into smaller more manageable pieces. It can also reduce bandwidth requirements as information needs vary greatly between processes.
- Compartmentalization with well-defined interfaces facilitates independent development and collaboration between disparate groups to develop the system.
- Specific consideration of context structure leads to more robust system solutions.

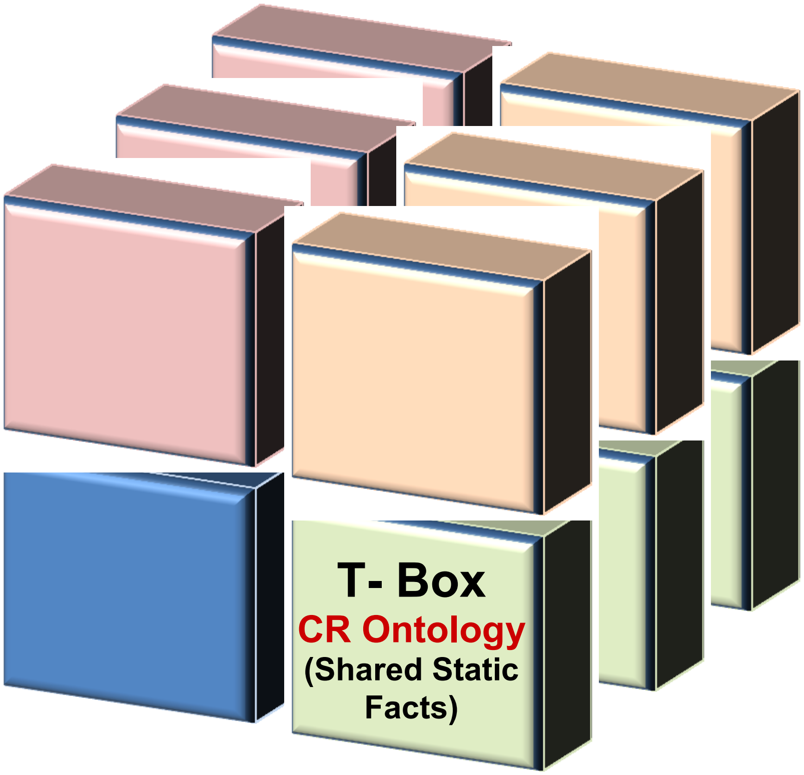


Figure 27: An overall communications context could be synthesized by piecing together schemas at different layers and functionalities. But this is easiest to maintain and extend when the schema are orthogonal.

Existing tools should be leveraged to the greatest extent possible.

A wide variety of components and tools have been developed in the area of inter-agent communications (e.g., XML, FIPA, OWL). Any IPA-based solution should look first to build on this rich ecosystem rather than custom designing the solution.

The Context Factory should be explored for further development

In discussions during the drafting of this document, we developed the concept of a *context factory*. The context factory is motivated by the need to support evolving, changing contexts. Thus a shared contextual solution (e.g., the T-Box in Figure 24) will need to evolve over time, and a CR will need to be able to construct and exchange context for communications and synthesize meaning from many different information sources, each with their own context. This function would be accomplished by the context factory, which, similar to a software factory, would be responsible for piecing together the different schemas needed to communicate with the associated sources.

In discussing this with members of the community, the context factory has also been analogized to an enterprise service bus which enables communication between disparate services. As a subtle difference, rather than acting as a bus between entities, the context factory would be a service available to each entity.

In a system with layered context where higher layers interpret lower layers, it will be difficult to support changes to lower layers in a way that does not break higher layers. To permit refactoring and reinterpreting of context via small changes to schema, the context factory will need a very good rule set and will likely need to maintain orthogonality at all levels.

Role for ontologies
Initially, we conceive of the schemas being segregated by application layer and / or information source type. To the extent that these schemas can be made orthogonal, this will be a relatively simple process. Other simplifications may be possible by allowing concept inheritance, e.g., a radio or network might inherit the user context and there could be layers of context. However, for further effectiveness, a CR will need to be synthesize and reason across disparate sources of information. Sometimes the same information will have different representations from different sources. Such a capability is typically enabled by ontologies, but these again must be scope limited. Such an example is MLM, which is proposed for use with 1900.5.

Use of Diffgrams, shared context, and non-human readable context for wireless communications
Due to the bandwidth constraints of the wireless medium, any information / context exchange mechanisms should at least consider bandwidth optimizations. Examples seen in this survey include the following.

- The use of a shared, pre-communicated context eliminates the need to transfer context at all during communications.
- Transmitting context in a side channel from the data
- Using terse, hierarchical dictionaries, e.g., as with CoT, can greatly reduce bandwidth consumption, but at the expense of human readability. But human readability is generally not needed at lower layers, perhaps only at the application layer.
- Contextual information need not be broadcast in human readable forms as done in XML. For example ASN.1 transmits contextual information in binary, bit-packed words to cut down on overhead. As needed at layer or contextual boundaries, routines can be deployed to convert between contextual representations.

7.3 Future Work

In the process of creating this document, we have identified several areas that merit further development.

Identify common contextual components for real-world applications
Several of the surveyed protocols shared the following contextual components in common:

- Who – about whom (or what process or agent) does the information apply, who generated the information, and who verifies that the information is valid
- When – the notion of time, e.g., when a message was generated, when is the information valid, was a common thread
- Where – what geographical region is this information applicable to

Identifying these common contextual components across domains into a well-defined list will then facilitate subsequent development of software processes to process information across contextual domains.

Explore the tradeoffs associated with a hierarchical / layered contextual framework

In practice, orthogonality at all levels / compartments and good rule sets to describe the use and interaction of contextual components will simplify subsequent refactoring and reinterpreting of context via small changes to schema. However, many cognitive radio applications (and information processing applications in general) will benefit from synthesizing awareness across contextual domains, so a development tension exists between ease of implementation / maintenance and developmental opportunities.

Contextual ARQ (Automatic Repeat Request)

In discussions during this project, we identified the need for both sender and receiver to maintain a consistent context but that as systems adapt and extend that contextual consistency may be temporarily lost. To address this possibility (eventuality?) a scheme could be developed whereby two agents are able to resynchronize their operating context. To develop such a capability the following questions need to be addressed:

- How can an agent recognize that a message or data or observed phenomenon is unfamiliar?
- Is there a general rule identifiable for handling conditions when an intelligent agent recognizes that it is operating in uncertain context?
- Who in the system should bear the responsibility of rectifying contextual deficiencies? The sender, receiver, or perhaps a third party database?
- Can we develop meaningful forward context correction when immediate context is lost in transmission?
- What are the steps involved with establishing the initial expected communications context?

Develop key components and tools for IPA communications

In the preceding section, we noted the need for a context factory and the support for contextual ARQ. Future work could more explicitly define these concepts and create sample implementations, which would facilitate further exploration of these concepts.

Appendix A: Acronym List

ACL	Agent Communication Language
API	Application Program Interface
A-SAP	Application SAP
CCEB	Combined Communications-Electronics Board
CE	Cognitive Engine
CHCSI	Chairman of the Joint Chiefs of Staff Instruction
COP	Common Operating Picture
CoT	Cursor on Target
CRO	Cognitive Radio Ontology
C-SAP	Communication SAP
DA	Data Archive
DISR	DoD Information technology Standards Registry
DoD	Department of Defense
DSA	Dynamic Spectrum Access
DTD	Document Type Definition
EIRP	Equivalent Isotropically Radiated Power
EL-CID	Equipment Location - Certification Information Database
ESC	Electronic Systems Center
EW	Electronic Warfare
EWCC	Electronic Warfare Coordination Cell
FCC	Federal Communications Commission
FIPA	Foundation for Intelligent Physical Agents
FM3TR	Future Multiband Multiwaveform Modular Tactical Radio
GCC	Geographic Combat Commander
GMF	Governmetn Master File
GNU	GNU is Not Unix
GPS	Global Positioning System
HQ	Headquarters
IPA	Internet Protocol
IPA	Information Process Architecture
IT	Information and Telecommunications systems
JFMO	Joint Frequency Management Office
JOA	Joint Operations Area
JRFL	Joint Restricted Frequency List
JTF	Joint Task Force
M2M	Machine to Machine
MAC	Medium Access Control
MLM	Modeling Language for Mobility
MoD	Ministry of Defense
MS	Monitor Service
M-SAP	Measurement SAP

NATO	North Atlantic Treaty Organization
NTIA	National Telecommunications and Information Administration
OSI	Open Systems Interconnect
OSMDD	Office of Spectrum Management Data Dictionary
OWL	Web Ontology Language
PFP	Partnership for Piece
PHY	Physical layer
QoS	Quality of Service
RELAX NG	REgular LAnguage for XML Next Generation
RF	Radio Frequency
SAP	Service Access Point
SCS	Spectrum Certification System
SDRF	Software Defined Radio Forum
SFAF	Standard Frequency Action Format
SGML	Standard Generalized Markup Language
SMADEF-XML	Spectrum Management Allied Data Exchange Format - XML
SNR	Signal-to-Noise Ratio
SSR	System Strategy Reasoner
SSRF	Standard Spectrum Resource Format
TCP	Transmission Control Protocol
TV	Television
TVBD	TV Band Device
TVWS	TV White Space
UAV	Unmanned Aerial Vehicle
UID	Unique Identifier
UML	Unified Modeling Language
US	United States
USRP	Universal Software Radio Peripheral
WG	Work Group
WGS	World Geodetic System
WinnF	Wireless Innovation Forum
WWW	World Wide Web
XML	eXtensible Markup Language