



Australian Government
Department of Defence
Defence Science and
Technology Organisation

Coalition Theatre Logistics (CTL) Advanced Concept Technology Demonstrator (ACTD) Web Services Documentation

Kuba Kabacinski¹ and Egon Kuster

**Command and Control Division
Information Sciences Laboratory**

DSTO-TN-0619

Abstract

This document describes the development and utilisation of web services in the Coalition Theatre Logistics Advanced Concept Technology Demonstrator (CTL ACTD) prototype. Included is a background on web services, issues with using web services, and technical details and specifications of all services used in CTL ACTD. The appendices contain sample Web Service Description Language (WSDL) files and the current XML Schema definition. Information contained herein captures current implementation details and can be used as a reference for future implementation. This document does not convey details of the operational CTL Spiral 1 and 2 currently deployed as part of JP2030's Movement Management System (MMS). Only details about the CTL ACTD prototype system are covered herein.

RELEASE LIMITATION

Approved for public release

¹ Kuba Kabacinski is a contractor from Consunet Pty Ltd working for DSTO Command and Control Division under the direction of Egon Kuster.

Published by

*DSTO Information Sciences Laboratory
PO Box 1500
Edinburgh South Australia 5111 Australia*

Telephone: (08) 8259 5555

Fax: (08) 8259 6567

© Commonwealth of Australia 2005

AR-013-356

March 2005

APPROVED FOR PUBLIC RELEASE

Coalition Theatre Logistics (CTL) Advanced Concept Technology Demonstrator (ACTD) Web Services Documentation

Executive Summary

The Coalition Theatre Logistics Advanced Concept Technology Demonstrator (CTL ACTD) produced a coalition environment for sharing logistics information with multiple partner nations in support of coalition operations. A core component of the CTL ACTD architecture is Web services, which provide the underlying communication infrastructure. These web services are key to transmitting logistics documents within the Coalition Information Environment (CIE).

During CTL ACTD development different issues and lessons were discovered; these are documented herein. This document also states a series of recommendations that CTL developers should/must follow for further development to overcome integration and development issues. As this document contains complete samples of the services' Web Service Description Language (WSDL) specifications and XML Schema definitions it can be used as both a reference to the CTL ACTD implementation or in future CTL development.

It should be noted that this document's contents do not detail any information about the CTL Interim Operational Capability Spirals 1 or 2 that have been operationally deployed and contained within JP2030's Movement Management System (MMS).

Contents

1	Introduction	1
1.1	Intended Audience	1
1.2	Document Structure	2
1.3	Revision Control	2
2	Target deployment environment	3
2.1	National Release Point	3
2.2	Coalition Server	4
2.3	Information Manager	4
3	Technology	5
3.1	XML Web Services	5
3.2	WSDL	6
3.3	Web Service Platforms	7
3.3.1	BEA WebLogic	7
3.3.2	Microsoft .Net	7
3.3.3	Sun Java Web Service Development Pack (JWSDP)	8
4	Data Types	9
5	Web Service Interoperation	12
5.1	Precedence	12
5.2	Interoperability	13
5.3	UDDI	13
5.4	RPC and Document Style Use	14
6	Web Service Security	16
7	Implementation Recommendations	18
8	Web Services	22
8.1	Movement Request – NRP	24
8.1.1	Interface Definition	24
8.2	Movement Request – Coalition Server	25
8.2.1	Interface Definition	25
8.3	Coalition Movement Requirement Plan – Coalition Server	28
8.3.1	Interface Definition	28
8.4	Organization – NRP	30
8.4.1	Interface Definition	30
8.5	Organization – Coalition Server	31
8.5.1	Interface Definition	31
8.6	Personnel – NRP	32
8.6.1	Interface Definition	32
8.7	Personnel – Coalition Server	33
8.7.1	Interface Definition	33
8.8	Equipment – NRP	34
8.8.1	Interface Definition	35
8.9	Equipment – Coalition Server	35
8.9.1	Interface Definition	35
8.10	Cargo In-Transit Visibility (ITV)	37
8.10.1	Interface Definition	37
8.11	Coalition ID – Coalition Server	39
8.11.1	Interface Definition	39
8.12	Language – Coalition Server	39
8.12.1	Interface Definition	39
8.13	Unit Conversion – Coalition Server	39

8.13.1	Interface Definition.....	40
8.14	Location – Coalition Server.....	40
8.14.1	Interface Definition.....	40
9	Future Directions	42
9.1	Authoritative Data Sources.....	42
9.2	Coalition IDs	42
9.3	Update XML Schema Definition	42
9.4	UDDI Registry Guidelines	42
9.5	Web Services Security	43
9.6	Emerging Web Service Standards.....	43
Appendix A:	Sample WSDLs for CTL ACTD Web Services	45
1 -	MovementRequestNRP	46
2 -	MovementRequestCS.....	49
3 -	MovementRequestServiceSOAP	53
4 -	CMRP	54
5 -	CMRPServiceSOAP.....	57
6 -	OrganizationNRP	59
7 -	OrganizationCS.....	61
8 -	OrganizationServiceSOAP	64
9 -	PersonnelNRP	65
10 -	PersonnelCS	67
11 -	PersonnelServiceSOAP.....	70
12 -	EquipmentNRP.....	71
13 -	EquipmentCS	73
14 -	EquipmentServiceSOAP.....	76
15 -	ITV	77
16 -	ITVServiceSOAP.....	81
17 -	CoalitionID	82
18 -	Language	83
19 -	Unit.....	84
20 -	Location	85
21 -	LocationServiceSOAP.....	87
Appendix B:	Coalition XML Schema Listing.....	89
References		137

Abbreviations

ACTD	Advanced Concept Technology Demonstrator
API	Application Programming Interface
CIE	Coalition Information Environment
CID	Coalition Information Database
CMRP	Coalition Movement Requirement Plan
CTL	Coalition Theatre Logistics
HTTP	Hyper Text Transfer Protocol
ICAO	International Civil Aviation Organisation
ITV	In-Transit Visibility
J2EE	Java 2 Enterprise Edition
JAXB	Java Architecture for XML Binding
JSP	Java Server Pages
JSTL	JavaService Pages Standard Tag Library
JWSDP	Sun Java Web Service Development Pack
MIEE	Multi-lateral Information Exchange Environment
MIME	Multipurpose Internet Mail Extensions
MMS	Movement Management System
NRP	National Release Point
NSN	NATO Stock Number
OO	Object Oriented
OS	Operating System
POD	Point of Disembarkation (entry location)
POE	Point of Embarkation (departure location)
RPC	Remote Procedure Call
SAML	Security Assertion Markup Language
SOAP	Simple Object Access Protocol
SSL	Secure Sockets Layer
UDDI	Universal Description, Discovery and Integration
UNLOC	United Nations Location code
UTM	Universal Transverse Mercator
VM	Virtual Machine
WSDL	Web Services Description Language
WS-I	Web Services Interoperability Organisation
WSS	Web Services Security
XML	eXtensible Markup Language

1 Introduction

This document is a design reference for all web services proposed to exist within the Coalition Theatre Logistics Advanced Concept Technology Demonstrator (CTL ACTD). As such it is used to validate the proposed design and as an implementation reference. Due to the proof of concept prototype nature of CTL ACTD and the rapid pace at which underlying technologies are evolving, this document is expected to change frequently, at least over the initial revisions. As such it should always be treated as a work in progress and proposed additions or changes should be directed to the authors or national CTL ACTD technical managers.

1.1 Intended Audience

The intended audience is designers and developers of the CTL ACTD project or others interested in applying similar ideas and techniques employed by CTL ACTD. It is assumed that the reader has a reasonable understanding of web services and related technologies. This document provides brief overviews on these topics; however, these are not sufficient to give in-depth understanding of all technological and implementation issues (see the references section for extra resources to learn about the underlying technologies).

It is highly recommended that the reader be familiar with the material outlined in the reference part of this document before reading the document body. However it is possible to read this document and consult the references for extended discussion on unfamiliar topics when encountered.

Developers familiar with the technologies and CTL ACTD architecture may use this document purely as a reference for implementing and designing future CTL ACTD web service interfaces. Section 8 herein provides such a reference and should be updated in the event new services are added. Additionally, Web Service Definition Language (WSDL) sample documentation is provided as an appendix to this document, although a full listing of WSDL documents will be available in the online Universal Description, Discovery and Integration (UDDI) registry in the implemented CTL ACTD architecture or on the internal DSTO SharePoint Portal [1].

This document does not cover details of the operational deployed version of CTL called CTL Spiral 1 or CTL Spiral 2² that is managed by JP2030 as part of the Movement Management System (MMS). Information contained herein is only relevant to the CTL ACTD prototype system; for more information about the operational CTL system please refer to the CTL Spiral 1 and CTL Spiral 2 documentation. It is likely that some of the web service definitions described herein could be implemented in a future operational version; therefore this document can be used as a reference for future CTL development.

² The Australian component of CTL Interim Operational Capability (IOC) Spiral 1 and 2 is called the "CTL Translation Service".

1.2 Document Structure

This document is structured to provide progressively greater detail. The general topic structure is as follows: introduction, environment, technology, data types, interoperation of components, security, recommendations, web service interface specification and appendices. The intended reading order is as presented although CTL developers can skip to the web service interface specifications and appendices.

The key sections are the CTL ACTD web services interface specifications and implementation recommendations. These sections, together with the WSDL sample files, represent the CTL ACTD web services design. All other sections can be treated as supplementary to facilitate better understanding of the design and deployment environment.

Accompanying this report is a CD-ROM that contains the Web Service Description Language (WSDL) interface and XML Schema definition files. If the CD-ROM is not present, contact Egon Kuster from Command and Control Division in the Defence Science and Technology Organisation.

1.3 Revision Control

The revision section at the beginning of this document is a record of this document's state. It documents how it has evolved and summarises changes made in each revision.

Because changes to the web service definitions may be made often and this may cause older versions to be incompatible, a 'since' version attribute is part of every web service definition in this document. This allows a reasonable means of determining how the web service definitions have changed over time. The version number is incremented every time an operation is added or changed in the web service definition. When only a service definition's wording is changed to clarify meaning the version number should remain the same; only when structure or variable and method names change should version numbers be updated.

2 Target deployment environment

In order to understand the target deployment environment, the CTL ACTD architecture needs to be understood. This architecture is documented in the CTL Architecture Overview [2] and Multi-lateral Information Exchange Environment (MIEE) Architecture [3] reports; however, for convenience it is summarised in Figure 1 CTL ACTD Architecture Overview below.

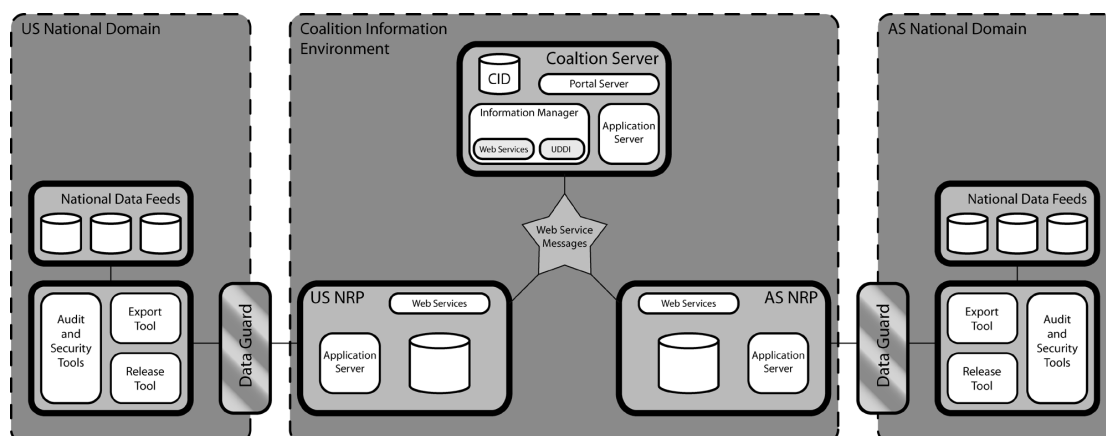


Figure 1 CTL ACTD Architecture Overview

National Release Points (NRPs) and the Coalition Server are the key components that utilise web services. The Coalition Server uses web services inside both applications running on the internal application server and within the Information Manager component. Coalition Server applications contain web service clients to send data requests or transmit data to web services in the Information Manager and NRPs.

2.1 National Release Point

NRPs use web services in applications running in their internal application server or to expose data contained in their NRP database. NRPs provide each nation with a conduit between national and coalition domains; therefore NRPs can be seen as a nation's electronic face in the coalition. On a programmatic level these faces are implemented as web services defined using WSDL. These web services allow an NRP to communicate with other nations (NRP-to-NRP communication) or to the Coalition Server. This web service communication is key to CTL ACTD information sharing and allows logistics data to be shared between multiple nations. Data transfer can occur as either 'pull' or 'push'; for example a nation could 'push' a movement request to another nation by sending a web service store message to another nation's NRP or the same data could be exposed to allow another nation's NRP to 'pull' and request the data.

Web services deployed in an NRP belong to the deploying nation. These services can be developed and run using any web service platform; however, developers must ensure that they are interoperable with the web services running in the Information Manager as a minimum interoperability requirement. Testing with other nations' NRP services should also be conducted to maximise interoperability within the

Coalition Information Environment (CIE). Section 3 herein details the technology options to develop NRP services.

2.2 Coalition Server

The Coalition Server implementation currently uses the BEA WebLogic platform to host a Portal Server, Application Server and the Information Manager. Thus web services deployed here must be implemented with BEA WebLogic Workshop, or as Java 2 Enterprise Edition (J2EE) applications that provide the same functionality and adhere to the core web service specifications³. Since future versions may use other J2EE application servers, vendor specific functionality should be kept to a minimum if possible.

2.3 Information Manager

The role of the Information Manager is to manage data moving around the CIE. Web services deployed here often query NRP web services to create a current representation of all or a subset of data in the CIE. The Information Manager also contains a set of coalition services that can be reused by other applications and services in the CIE. An example Information Manager service could be a translation service that allowed any application or NRP to translate data between languages. By locating such coalition services in the Information Manager they are not duplicated on each NRP and provide additional services that CTL ACTD application and service developers can utilise.

A UDDI registry to store CIE web service descriptions is hosted in the Information Manager. All web services (NRP and Coalition Server web services) must register themselves with the UDDI so that they are visible to coalition applications and services. This UDDI server is the authoritative source of web service locations in the CIE.

³ The core web service specifications are SOAP, WSDL and UDDI. Each is described further in Section 3.

3 Technology

The key technology used by CTL ACTD is web services. Web services are programmatic interfaces that exist in a distributed environment, such as the Internet or in this case the coalition network, that allow two or more components to communicate. Web services accept requests, perform their function based on that request, and may return a response. The request and response can be part of the same transaction (synchronous), or can occur separately (asynchronous) where the requestor does not wait for a response⁴. Both request and response usually take the form of eXtensible Markup Language (XML) data, a portable data-interchange format, and are delivered over a wire protocol, such as Hyper Text Transfer Protocol (HTTP).

Web service transactions are usually conducted between businesses. A business that is a provider of a service can also be a consumer of another service. Likewise, some CTL ACTD web services provide functionality that operates as a web service client.

The Web Service Interoperability Organization (WS-I) [5] works to standardise web services and ensure interoperability between vendor implementations. WS-I provides guidance, best practices and other resources for developing web services and ensuring interoperability. A key outcome produced by WS-I is the Basic Profile 1.0 [6] specification. This specification clarifies and removes ambiguity in the collection of standards that constitute XML web service technology. It is expected that the WS-I efforts in standardisation will continue and new revisions of the Basic Profile and other profiles will become available over time.

To minimise interoperability problems in the CTL ACTD architecture, adhering to standards for implementing web services is of outmost importance. Interoperability problems may materialise through the use of non-homogeneous technologies to implement individual architectural components. It is one of the CTL ACTD design goals to allow for multiple web service technology implementations and vendors to be used, as this allows greater flexibility in terms of development. It also allows for vendor specific features and developer expertise in any web service technology to be leveraged. As long as the external web services communication adheres to the WS-I Basic Profile specification these different implementations will operate harmoniously together.

Web service technology encompasses multiple standards including Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI). These standards and components are discussed further herein.

3.1 XML Web Services

XML web services are based on the SOAP specification. SOAP is “a stateless, one way, XML based information exchange paradigm” [7]. Applications can build on top

⁴ A call back function can be used to receive feedback at a later time as part of a second transaction.

of SOAP to provide various communication types that can maintain state and have bi-directional messaging. SOAP is used to create SOAP Messages that in turn are used in SOAP Message Exchanges to perform transactions. SOAP Messages are not limited to textual data, as XML content can include binary content in the form of Multipurpose Internet Mail Extensions (MIME)⁵ encoded elements. Further, the SOAP standard itself provides for XML messages with attachments that allow for binary data exchange. Nevertheless, binary data exchanges should be kept to a minimum due to interoperability issues with binary data messaging.

Recommendation: Binary data exchanges should not be used.

There are two styles of SOAP operations: Remote Procedure Call (RPC) and Document. RPC style operation messages contain parameters or return values, while Document style messages contain XML documents. When Document style is used the web service's WSDL specifies the document structure used in operations. RPC style only requires this additional definition if complex types are used in the operation request or return parameters. Base types are automatically bound to the underlying language that implements the RPC style web service. RPC style web services are generally used when applications require a simple interface, while Document style services allow for larger and more complex data types to be exchanged. More detail and guidelines on using RPC and Document style services can be found in Section 5.4. The web service interface definitions are described using WSDL.

Recommendation: SHOULD use RPC literal style web services for operations that use only basic data types.

Recommendation: MUST use document literal style for operations that use coalition data types and other complex data structures.

3.2 WSDL

"WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints (services). WSDL is extensible to allow description of endpoints and their messages regardless of what message formats or network protocols are used to communicate, however, the only bindings described in this document describe how to use WSDL in conjunction with SOAP 1.1, HTTP GET/POST, and MIME."[8]

Additional clarification to the WSDL specification is documented in WS-I Basic Profile 1.0 specification – this removes some of the ambiguity that the original specification contains. CTL ACTD uses the WSDL 1.1 specification; however, as new specifications continue to evolve their applicability should be reviewed periodically for inclusion. It should be noted it is likely that future versions of WSDL

⁵ Microsoft uses an alternative encoding technique called DIME that is incompatible with MIME. CTL ACTD currently does not use binary data so the incompatibilities with encoding techniques are not an issue.

specifications may not be backwards compatible and vendor support always lags behind the development of specifications, making early adoption of new versions difficult.

Recommendation: Future web service standards (SOAP, UDDI, WSDL) should be reviewed for applicability in CTL ACTD.

3.3 Web Service Platforms

The CTL ACTD architecture allows for any web service platform to be used. This design goal allows participants of CTL ACTD to choose a platform that suits their needs, but still produces interoperable functionality. To maximise functionality reuse, components developed for CTL ACTD should be platform independent where possible so that they can be deployed on any compatible vendor platform. An example is a component of the Coalition Server that has functionality reusable in an NRP. If such a component is platform independent, then it could be deployed in the NRP without redeveloping the same functionality for different platforms. This minimises the overall effort needed to implement CTL ACTD functionality. Some web service platforms offer such portability, and this is generally true for those that implement the J2EE specification.

Recommendation: SHOULD develop components that are platform independent.

The initial prototype stage of CTL ACTD considered a small sample of established web service technology vendors. The products employed were BEA WebLogic, Microsoft .NET and Sun Microsystems Java Web Services Development Pack (JWSDP).

3.3.1 BEA WebLogic

The BEA WebLogic Platform implements and extends the Sun J2EE platform specification. The main components are Workshop, Portal, Integration, Application Server and a server-side optimised Java Virtual Machine (VM) called JRockit.

In the CTL ACTD context, the BEA WebLogic 8.1 platform was used to develop and deploy web services and portal based web applications for the Coalition Server. BEA Workshop allows for rapid development of components; although these components use J2EE technology, they may not be easily ported to other J2EE compliant application servers due to the use of BEA WebLogic specific extensions. This and other interoperability issues make the BEA WebLogic 8.1 Platform difficult to use for developing web services. Nevertheless, such a claim can be made for most J2EE implementations due to vendor specific features introduced for market advantage and differentiation purposes.

3.3.2 Microsoft .Net

The Microsoft .NET framework is used for building and running many types of software, including web-based applications, smart client applications, and XML web services.

Unlike the other two J2EE-based web service platforms⁶ used by CTL ACTD, the .NET Framework is owned and maintained by Microsoft and relies heavily on the Microsoft Windows platform. Due to its high integration with the underlying Operating System (OS) it offers potentially higher performance on equivalent hardware and arguably easier maintenance. However, it sometimes does not implement industry standards, instead developing proprietary 'standards' to attain advantage over other platforms, resulting in components that are more difficult to integrate in a heterogeneous environment such as CTL ACTD.

3.3.3 Sun Java Web Service Development Pack (JWSDP)

The JWSDP is an integrated toolkit that allows Java developers to build, test and deploy XML applications, web services, and web applications using the latest web services technologies and standards. Technologies in JWSDP include the Java Application Programming Interfaces (APIs) for XML, Java Architecture for XML Binding (JAXB), JavaServer Faces, WS-I sample application, XML and Web Services Security (WSS), JavaServer Pages Standard Tag Library (JSTL), Java WSDL Registry Server, Ant Build Tool, and Apache Tomcat⁷ container.

JWSDP is the most open web services platform used in CTL ACTD. It also provides the least development support tools as it is a development kit rather than a production platform. JWSDP is developed through a community process and uses several components that have been developed by the open source Apache Foundation. This platform is only intended to provide support for cutting edge web service technologies, unlike BEA WebLogic and Microsoft .NET. This results in only a subset of J2EE technologies being provided; however, they are implementations using the latest specifications and indicative of future J2EE products. Recently the JWSDP technologies have become part of the J2EE 1.4 specification; therefore vendors that implement the J2EE platform will eventually provide the same functionality available in JWSDP now.

⁶ BEA WebLogic Platform and Java Web Service Development Pack (JWSDP)

⁷ Java Servlet and Java Server Page (JSP) container

4 Data Types

CTL ACTD web services manipulate data that can be partitioned into three groups: CID, NRP and CIE data.

CID data is any data located in the Coalition Information Database (CID) on the Coalition Server. CID data is shared by all participants and is stored and accessed using the coalition data formats. Data can be uploaded to the CID directly from a Nation's NRP or by using a coalition application that can access the Coalition Server's web services to store data.

Recommendation: CID data is stored in the CID database on the Coalition Server and MUST only use coalition data formats.

NRP data is any data located in any nation's NRP within the CIE. Participating nations create this data and expose it using coalition data formats and web services on the NRP. Internally to each NRP, data may be stored and manipulated using any data formats. National data formats should never be exposed to the CIE, as they are meaningless outside of the national domain; the NRP forms a single gateway to the national domain. A nation may use national data formats and interfaces to connect with national systems if required, although these should never be exposed to other NRPs, coalition applications or the Coalition Server. National NRP interfaces are to be defined in NRP documentation and are a national issue.

Recommendation: NRP data is stored on each NRP in the CIE and exposes national data using coalition data formats. All external interfaces to the CIE on the NRP MUST use coalition data formats.

CIE data is any data located in the Coalition Information Environment (see Figure 1 page 3). This encompasses CID data, NRP data and any other data that can be accessed or stored in the CIE. Any data type can fall into this category; however, only coalition data formats are used and defined by this document. There are no restrictions on what data the NRP components use internally, providing that they do not attempt to use non-coalition formats to send data to the Coalition Server or other nations' NRPs. Only CTL ACTD interfaces that make use of the Coalition data formats are defined in this document.

Recommendation: Only coalition web service interfaces are to be defined in this document. National internal NRP interfaces are to be defined by the NRP owner in the NRP documentation as these only affect national requirements.

The coalition data types are described using the XML Schema standard as defined in XML Schema Part 1: Structures [9] and XML Schema Part 2: Datatypes [10]. The Coalition Schema is an evolving definition. Initial versions brought large changes to the definition; however, future versions should only have minor changes to fix bugs found during development and use. It is recommended that the Coalition Schema

changes are kept to a minimum and not changed regularly as it can create an unproductive development churn⁸.

In this document the data types defined in the Coalition XML schema are referenced using a 'cool:' prefix or namespace name to use the XML terminology. For instance 'cool:IdT' references a data type called 'IdT' in the coalition schema. For full documentation of coalition data types please consult the Coalition Schema definition in Appendix B: Coalition XML Schema Listing.

In addition to the custom coalition data types, the data types in the XML Schema standard can be used. These standard types are the basic types commonly used in all computing environments such as character and numeric values. The full listing of built in data types is defined in the XML Schema Part 2: Datatypes [11] document.

Figure 2 summarises these built-in types; they are referred to by name herein, for instance a 'string' type.

⁸ Development churn can happen when schema changes require corresponding changes in the software. If this happens frequently, developers will spend all their resources satisfying changes rather than implementing new features, fixing errors or enhancing capability.

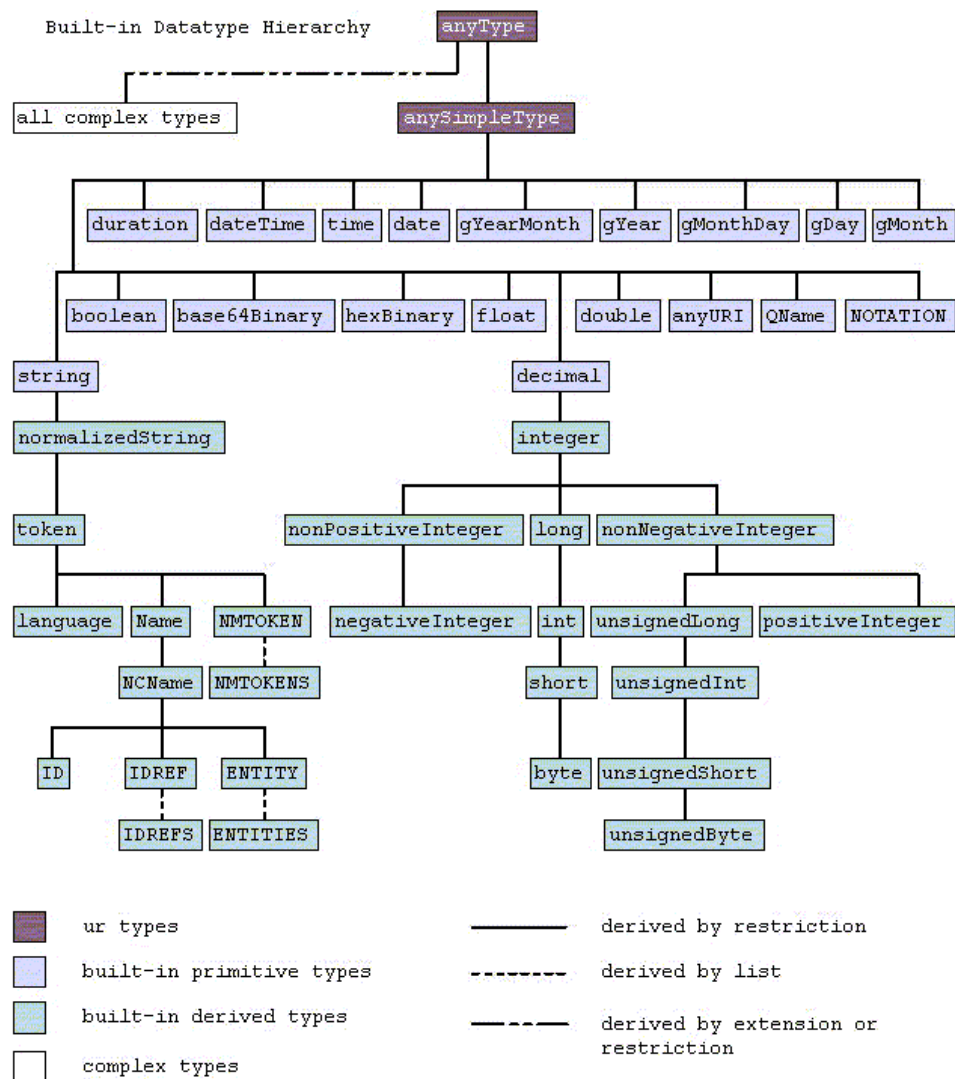


Figure 2 Built-in XML Schema Data types

5 Web Service Interoperation

In the CIE, all interactions between components are performed using SOAP web services over HTTP. The services use coalition data types to exchange data and perform actions. Services may use asynchronous or synchronous message exchange types. The protocol level and business process level exchange styles are not interdependent, making it possible to implement asynchronous business processing using synchronous message exchanges with SOAP. There is no constraint on the processing architecture in the CIE and it is expected that both synchronous and asynchronous processes will be utilised.

CTL ACTD web services can function as standalone services or can provide web service chaining functionality. A standalone service does not require other CTL ACTD components to perform its function. Chained services can result in multiple web service transactions from a single web service call from the client. This allows for maximum functionality reuse and can ensure better data integrity for the CIE through verification with authoritative data sources. For instance, a Coalition Server web service may check the original NRP data source for updates before returning cached data; this transaction has chained the NRP call to the original Coalition Server request. Web service chaining can have a detrimental effect on performance as each subsequent web service call results in additional delay. Most chained web services reside in the Information Manager; however, this does not preclude deploying chained services in other places in the CIE.

5.1 Precedence

When data is retrieved, rules of web service precedence must be followed to ensure that the data is up to date and valid. This is due to multiple data paths that can be followed by information. A significant proportion of CIE data is created by NRPs and this data could be pushed to the Coalition Server or just exposed by the NRP. The Information Manager and coalition applications may also generate CIE data. In some situations it is valid for an NRP to 'push' data directly to another NRP for nation-to-nation communication. This freedom of data exchange makes it difficult to establish the authoritative data source for a particular data item⁹.

In order to resolve the authoritative data source issue, the general rule applied is that the Coalition Server and the Information Manager are always the authoritative sources of data. Thus when no special case for sourcing data from a specific NRP exists, the data must be sourced from the Coalition Server and its web services. It is then the responsibility of the Coalition Server to take an appropriate action to determine if the data is up to date; initialising a distributed query using web service chaining to all NRPs in the CIE may achieve this. This distributed query will search all potential data sources and determine the current data instance. This centralised approach simplifies the implementation of individual NRPs as no special functionality needs to be implemented to guarantee that data is up to date.

⁹ Further research needs to be completed on handling authoritative data sources in distributed environments such as the CIE.

Recommendation: The Coalition Server and Information Manager SHOULD be considered as authoritative source unless updated data is available at an NRP.

The CTL ACTD architecture does not provide a means to record origin and authoritative data source metadata. This has been identified as a problem and should be addressed in future revisions.

5.2 Interoperability

Use of different types of platforms to implement CTL ACTD web services requires interoperability between components to be managed. For the purpose of this document, it is assumed that web services created on the same platform will not incur interoperability issues. However lessons learnt from development in CTL ACTD has shown that different web service platforms do not interoperate well, especially with complex web services.

The coalition data types used are quite complex when compared to web service examples provided by different platforms. This high complexity often causes problems when creating platform language bindings to the XML data types. Specific problems were caused by the use of collection data structures in the XML schemas.

Interoperability issues are further amplified by different interpretations of various XML web service specifications. This problem is caused by ambiguous definitions, which in turn have been interpreted differently by platform vendors. Such differences make it quite difficult to implement a web service that is compatible with all platforms. To enable interoperability, all participating platforms need to be analysed and the differences identified. Fortunately, this problem has been rectified by many organisations in the industry and action has been taken to create a less ambiguous standard. The product produced from this work is the WS-I Basic Profile 1.0 standard [6]. All CTL ACTD web services must adhere to WS-I Basic Profile 1.0. By adhering to this recommendation, CTL web services should be interoperable despite implementation and platform differences.

Recommendation: All web services MUST adhere to the WS-I Basic Profile 1.0 standard.

5.3 UDDI

With the exception of the Coalition Server and its web services, the CIE environment can change as servers appear and disappear as nations enter or leave the coalition. Changes can also occur with the deployment of new or updated NRP and Coalition Server software. Therefore a method to discover available web services and their location is required. For this reason all web services are registered in a common UDDI registry embedded in the Information Manager. Clients in the CIE discover available services or search for required services by querying this UDDI registry.

Information Manager functionality is heavily dependent on UDDI as it allows for tracking of available NRPs and their components. This is essential for Information Manager web services that collect data from the entire CIE and provide service

chaining functionality. Therefore NRPs not registered in the UDDI will not be queried by the Information Manager services.

Recommendation: All CIE services SHOULD be registered in the CTL ACTD UDDI registry.

Whenever a new web service is deployed in the CIE it should register itself with the Information Manager's UDDI. This is especially important for NRP components when first connected to the coalition network. Without their presence in the UDDI, only components explicitly 'aware' of their existence can use them; this is not the intended usage as the Information Manager component can't access NRP information automatically and therefore the CIE will behave as if the new NRP did not exist. Likewise, when web services are no longer available they should be de-registered from the UDDI registry. This ensures correct functionality by not needlessly forcing components to search for web services that are non-existent now. This helps maintain optimal performance, as components do not wait for timeouts on non-existing resources. Services not utilised by other nations are not required to register with the UDDI; however, this is the exception to the general rule of registering all web services.

Recommendation: All CIE services SHOULD be de-registered from CTL ACTD UDDI when removed from the CIE.

To clarify this further, imagine a situation where a new NRP with multiple web services is deployed as part of a new nation joining the coalition force. Each web service in the NRP that implements a part of the CTL ACTD architecture must be registered in the Information Manager's UDDI. The NRP may also contain services used by internal national tools that do not interact with the Coalition Server nor other NRPs. These national web services do not need to be registered with the UDDI. Registering such web services is not prohibited, so the Information Manager's UDDI registry can be used for all web services if required.

5.4 RPC and Document Style Use

SOAP-based web services offer two operation styles, RPC and Document. RPC style services use operations that contain zero or many input parameters and may return zero or many response parameters. Document style services use XML documents as the service's input and output.

The key difference between the two styles is that in RPC style services the parameters are serialised to XML by the underlying web service framework, whereas in Document style services the programming logic deals with XML documents directly. It may seem that RPC style exchanges are simpler to develop, as the programmer does not handle the XML directly. This holds true for operations that have simple parameter and return types and the underlying XML nature of web services can be completely hidden from the application developer. However when the data types used are complex, difficulties arise in using the RPC approach, as complex bindings

may be required and the differences in implementation languages to XML bindings may produce differing results on different platforms.

Using Document literal style services, programmers must handle the XML in the message directly. No support is provided by the web service framework other than extracting the XML body from the SOAP message, unlike RPC style services. This provides developers with the opportunity to validate XML content against an XML schema and to use custom language bindings. Most web service frameworks handle both document and RPC style services in a similar way from a programmatic viewpoint. However, Document style services provide more flexibility, with few or no negative consequences apart from additional effort by the programmer to parse and use the XML data.

In CTL ACTD most request and return parameters use Coalition XML schema types. These types can be complex and utilise recursive structures, which can create problems with some XML schema tools even though all schema definitions are well-formed and valid XML schema. Additionally, the Coalition XML schema is specified using multiple schema definition files. The referencing of documents in XML schema files can be problematic as some platforms have difficulty with relative links. Absolute links cannot be used as CTL ACTD is developed, tested and deployed on different networks/platforms/machines, often with no Internet access, making absolute document references difficult to maintain. If a platform does not support relative links then the developer will be required to manually change all schema links to absolute links by manually editing each schema definition file.

Recommendation: Relative links between XML Schemas SHOULD be used.

Recommendation: Web service platforms chosen MUST support recursive XML Schema structures and relative referencing.

The general rule when choosing between RPC and Document style exchanges in creating new web service definitions is that unless the operation parameters are only basic types, as identified in Section 4 (see page 9), Document style exchanges should be used. This should exclude most Coalition XML schema data types from being used as RPC style parameters because of their complex nature. In most cases this is a more efficient way to implement the web services required, as the Coalition document processing logic is developed as part of tools rather than the web service that exposes them.

Recommendation: Document style web services SHOULD be used in all cases except when the operation parameters are ONLY using the basic types as defined in Section 4.

In terms of performance, RPC style web services often result in poorer performance than Document literal services. This occurs because XML message parsing cannot be skipped or delegated to an optimised implementation, as is the case with Document style services. However, delegating XML processing to the developers carries the risks of introducing errors or inefficiencies due to poor implementation and non-standard parsing conventions.

6 Web Service Security

The CTL ACTD architecture is designed for deployment on private networks only accessible by coalition participants. Due to this, the CTL ACTD security needs are significantly different from those usually identified by organisations using public networks (eg. Internet). For instance, the trust level for participants should be roughly equivalent, as all participants are operating within an agreed coalition force. CTL ACTD design also assumes that security mechanisms are in place to secure communication channels between national and coalition networks and any systems on the network itself.

However, it is envisaged that in the future different levels of trust and data access may be required. To facilitate this, access control needs to be implemented at the data level rather than just the interface level supported by the current CTL ACTD architecture. Web service security standards are not limited to provision of communication confidentiality but also provide functionality for ensuring integrity of data and verifying message source to name just a few.

The current CTL ACTD architecture does not mandate the use of transmission confidentiality, authentication and integrity checking for the Coalition Server or NRPs. However, use of web service security features is not restricted, given that NRP's implementations are free to include additional services to those described in this document. Therefore, secure services or operations may be implemented but will not be used by the current core CTL ACTD functionality¹⁰.

The last point of national data access control before release into the CIE from a coalition participant is the NRP. Releasing data into the CIE is not "undoable" and therefore the releasing participant must be certain that all other participants are allowed to view the data. It is up to the NRP and Data Guard designs and implementations to ensure that sensitive national data is not unintentionally released into the CIE. This generally implies that only coalition releasable data is put into an NRP and, more correctly, onto the coalition network, as from a security point of view all data on the coalition network must be releasable to all other coalition members. However, data released into the coalition and exposed by an NRP may have discretionary access controls applied that only let limited CIE users/applications view or manipulate the data. The current version of the CTL ACTD architecture does not fully define how discretionary access controls are to be applied and will need to be further developed in future revisions.

Although outside of the CTL ACTD scope, the NRPs could be used to conduct direct information exchange without publishing to the CIE. Since it is permissible for NRPs to implement new services and there is an underlying shared network, it is not unreasonable that some participants may choose to transport data directly and confidentially between national NRPs. In such cases, the NRP implementation needs to ensure that appropriate security measures are put in place and web service security data confidentiality features are used. Since CTL ACTD web services make use of HTTP for transport, the basic data integrity and access control are provided at

¹⁰ As the web services security standards mature these will be incorporated into future versions of the CTL ACTD architecture.

the transport level. HTTP already allows secure communication by using Secure Sockets Layer (SSL), but such security measures most likely would not meet the required security levels.

The main source of web service security specification is the OASIS Web Services Security (WSS) Technical Committee [12]. This is a consortium of industry representatives who collaboratively develop standards based on industry best practice and needs of the organisations they represent. Similarly to the core web service standards, different vendors can implement the WSS standards differently. To maintain interoperability with WSS products, WS-I Organization has developed the Basic Security Profile [13]. This profile addresses transport layer security, SOAP message security, use of security tokens, signing and encrypting XML, and algorithms used.

Recommendation: All security implementations MUST adhere to WS-I Basic Security Profile 1.0.

The OASIS Technical Committee is working on specifications that outline how message security can be achieved and has produced the SOAP Message Security 1.0 standard [14]. The OASIS committee has also produced Username Token Profile 1.0 [15] and X.509 Token Profile 1.0 [16] standards that describe how to use the UsernameToken with the WSS specification and how to use X.509 Certificates with the WSS SOAP Message Security specification. Security Assertion Markup Language (SAML) is yet another OASIS output in the WSS space. SAML provides a mechanism for secure communication of identity information.

The adoption of any security features will be limited by vendor support, which is likely to lag behind standardisation processes.

7 Implementation Recommendations

The following recommendations are included to ensure that all web services in the CTL ACTD environment can interoperate easily. The recommendations are of two types: a strict MUST and a preferred but not required SHOULD. Additionally a NOT is used to indicate a negation of either type. Some are duplications of recommendations documented earlier.

Recommendation 1:

MUST adhere to WS-I Basic Profile 1.0.

Reason:

Ensure interoperability of web service implementations across different platforms. Not all web service platforms adhere to this recommendation by default as it is reasonably new. This recommendation should be supported by all future web service platforms from all vendors.

Recommendation 2:

MUST adhere to WS-I Security Basic Profile 1.0

Reason:

Ensure interoperability of web service security features across platforms, if security features are being used. Different platforms may not adhere to this recommendation by default.

Recommendation 3:

MUST use document literal style for operations that use coalition data types and other complex data structures.

Reason:

The **SOAPBody** of web service messages can be validated against the coalition schema before any further processing begins. This can be done by the web service platform by enabling a strict validation option.

During an RPC data exchange the XML data is serialized into base types. The coalition data types are quite complex and can cause some XML serialization engines to fail. This is especially the case where a web service is discovered and accessed at run time.

More information can be found in Section 5.4 – RPC and Document Style Use – on page 14.

Recommendation 4:

SHOULD use RPC literal style for operations that use only basic data types.

Reason:

From an implementation point of view, it is generally easier to use RPC style operations when ONLY basic data types are used as input and return parameters.

Recommendation 5:

MUST NOT use `encodingStyle` attributes on `SOAPBody` content. Therefore only literal style is allowed.

Reason:

This is a WS-I Basic Profile 1.0 recommendation. Encoded message content also causes an additional processing requirement on both transaction ends. This creates an adverse performance effect that may be evident during heavy use.

Recommendation 6:

SHOULD define custom elements for a web service in a separate XML schema file.

Reason:

By not embedding data type and element definitions in a WSDL document they can be easily reused in other WSDL documents.

It is often practical to create custom bindings of those XML types to a programming language. The tools that do this always process XML schema documents, but rarely extract XML schemas from WSDL documents.

For platforms that have problems using the custom definitions referring to coalition data types, these definitions can be substituted with ones that specify XML “any type” elements. In such cases, the web service platform will not attempt to perform any XML processing and it will be up to the further processing logic to deal with the platform specific problems.

Recommendation 7:

SHOULD use relative links to XML Schema documents in WSDL documents.

Reason:

The use of Relative URLs is flagged as not sufficiently specified by the Basic Profile 1.0 recommendation. The CTL ACTD development environment makes use of absolute linking prohibitive, due to the difficulty of enforcing a single source, either by name or physical location, whereas XML Schemas could be accessed at all times by all participants.

Recommendation 8:

SHOULD use request/response message exchange types.

Reason:

Allows for higher level of error control and operation verification than unidirectional operations. However, where network resources need to be conserved this may not be appropriate.

Recommendation 9:

All CIE services SHOULD be registered in the CTL ACTD Information Manager UDDI registry.

Reason:

So new web services can be found by all CIE users and applications.

Recommendation 10:

All CIE services SHOULD be de-registered from CTL ACTD Information Manager UDDI registry when removed from the CIE.

Reason:

So old functionality is not falsely advertised as available.

Recommendation 11:

SHOULD implement all operations of a web service interface definition. If not implementing functionality for all operations, those with unimplemented functionality MUST return a SOAP Fault report stating that the operation is not implemented.

Reason:

The interface of a web service will always be complete so unexpected exceptions thrown by web service frameworks (such as operation not found, etc.) will not be thrown. This way partial implementation of web services can be managed in a consistent manner throughout the system.

Recommendation 12:

Binary data exchanges SHOULD NOT be used.

Reason:

Due to two conflicting standards for binary attachments to SOAP messages, MIME and DIME, interoperability may not be possible. If required, binary content can be encoded into the body of the SOAP message using standard XML Schema data types, although performance is lower than using SOAP Messaging with attachments.

Recommendation 13:

Future web service standards (SOAP, UDDI, WSDL) SHOULD be reviewed for application in CTL ACTD.

Reason:

Because the standards underpinning CTL ACTD will continue to evolve, it is essential that they are reviewed periodically to ensure that new implementations can use products available and enhanced features are leveraged. It must be noted that vendor support may initially be limited and must be taken into account when selecting new standards.

Recommendation 14:

SHOULD develop components that are platform independent.

Reason:

Application servers that support the J2EE specifications provide this functionality. Enables a single implementation to be deployed on different platforms and reduces the cost of implementing and deploying CTL ACTD components.

Recommendation 15:

CID data is stored in the CID database on the Coalition Server and MUST use only coalition data formats.

Reason:

The use of only coalition data formats ensures that tools developed will understand and utilise data contained within the CID.

Recommendation 16:

NRP data is stored on each NRP and exposes national data using coalition data formats. All external interfaces to the CIE on the NRP MUST use coalition data formats.

Reason:

The use of coalition data formats ensures that tools developed can use and understand exposed NRP data.

Recommendation 17:

The Coalition Server and Information Manager SHOULD be assumed as authoritative sources unless updated data is available at an NRP.

Reason:

As the current CTL ACTD architecture contains no mechanism for labelling data with an explicit authoritative source, this assumption simplifies the issue of authoritative data sources.

Recommendation 18:

Selected web service platforms MUST support recursive XML Schema structures and relative referencing.

Reason:

Recursive XML schema structures and relative referencing exist in the current CTL ACTD Schema and WSDL definitions. Web service platforms chosen must be able to use these otherwise problems will be encountered during development.

Recommendation 19:

Only coalition web service interfaces SHOULD be defined in this document. The NRP owner in the NRP documentation SHOULD define national internal NRP interfaces as these only affect national requirements.

Reason:

All web services other than the core coalition web services are out of scope for CTL ACTD.

8 Web Services

CTL ACTD web services provide the communication layer, which is the core functionality, to enable coalition interoperability. The interfaces are defined so that new components¹¹ can be developed and plugged into existing infrastructure without the need to have knowledge about the current CTL ACTD system state.

The web services definitions provided below identify the functionality and input/output parameters and should be used for implementing the functionality behind the coalition web services. For final definitions of the web service interfaces consult the WSDL files. If a web service environment supports the 'WSDL first' web service development model, these WSDL definitions can be used to generate stub and client code automatically, allowing business logic to be plugged in easily and reducing development time.

Another method of developing web services is by creating the functionality, in an Object Oriented (OO) environment usually by creating a class or classes that implement it, and then infer the web service from this. This approach is best used when developing RPC style web services; however, with more complex structures differences can arise in the resulting web service definition. This difference is due to factors such as platform specific conventions and a lack of support for the WS-I Basic Profile 1.0 specification. If web services in the CTL ACTD were developed in this ad-hoc way, the result would more than likely not function with multiple web service platforms. By developing from a common WSDL, interoperability issues can be minimised.

¹¹ A new nation joining the coalition and deploying their National Release Point with its web services is one example of a new service.

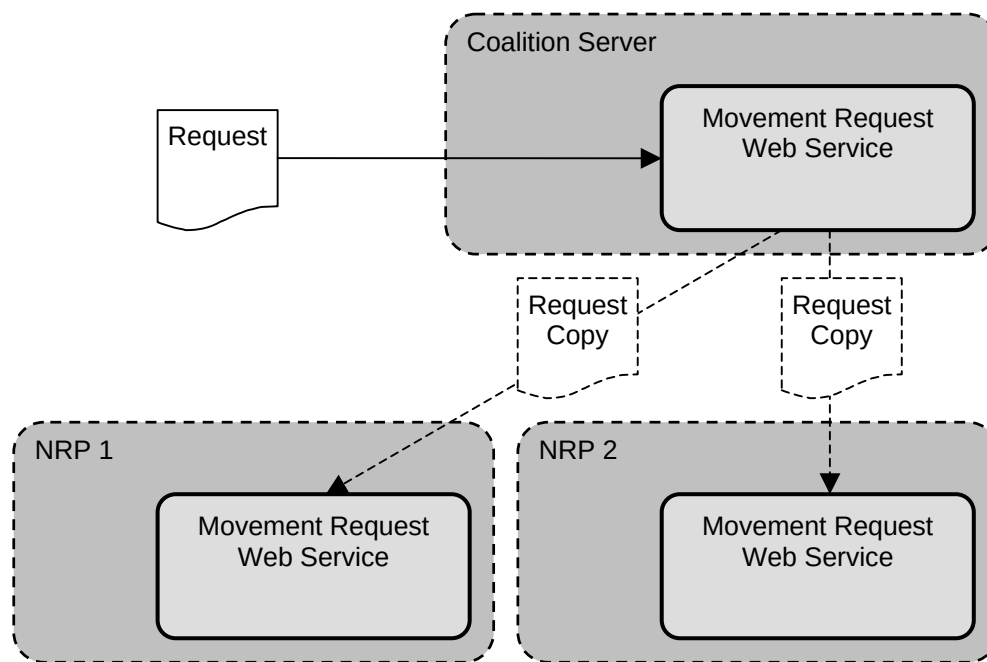


Figure 3 Web Service chaining example

There are two locations where web services can be deployed – Coalition Server or NRPs. In most cases web services that exist on both the Coalition Server and NRP have similar functionality. An example of this is the Movement Request web service, where many methods take the same arguments; however, the functionality differs slightly. This is intentional as it makes chaining of web service requests easier by passing arguments from the original request to chained requests and collating the responses, as shown in Figure 3. Services that have the “FromAny” suffix in their name make use of this chaining functionality. When such a service is queried, it will query itself and all other sources providing the required information, resulting in the Coalition server and all registered NRPs being searched.

Web service errors are handled using web service faults. SOAP fault messages are a mechanism similar to that of an exception used in some programming languages, and provide a standard way of reporting unexpected execution errors. The delivery of faults to the caller depends on the message transfer paradigm – for instance in a request-response exchange, the response may carry the fault. In unidirectional operations fault reporting may not be directly possible. CTL ACTD web services should return fault messages as specified in their definition or at any time unexpected conditions prevent the service from returning the required result. A fault message may also be sent as a way of signalling that an operation is taking too long or that the expected quality of service is not going to be met¹².

The rest of this section defines the interfaces used. Each interface definition identifies whether it is for an NRP service or a Coalition Server service. Coalition Server services gather data from both the Coalition Server’s database and the NRPs. Data in

¹² However this functionality is currently not used in CTL ACTD.

an NRP is accessed via the NRP web services, which is also the method used by Coalition Server services to extract NRP data. For comprehensive definitions of each service please refer to the WSDL files contained in the CTL ACTD Information Manager's UDDI registry within the live implementation. Sample WSDL files are provided in Appendix A on page 45 for completeness.

8.1 Movement Request – NRP

Description:

Provides access and management functionality for national movement requests. This Movement Request web service is deployed in NRPs. The data it references is owned by the NRP and its actions only affect the NRP it is deployed within.

Dependencies: none

Web Service Style: Document

8.1.1 Interface Definition

getAllMovementRequests	method
Retrieves all movement requests available in the NRP that this web service is running within. Returns a MoveRequestList that has 0 or more movement requests.	description
none	arguments
cool:MoveRequestList rtmList	return
0.1, removed 0.9	since ver.

getMovementRequest	method
Retrieves a single movement request identified by the id (identification label) and ver (version) arguments from this NRP service. If no movement request is found, a fault message is returned.	description
cool:IdT id, cool:VersionT version	arguments
cool:RequestToMoveT rtmList	return
0.3	since ver.

removeMovementRequest	method
Removes the movement request identified by the id (identification label) and ver (version) arguments from this NRP. Returns true if successful. If specified movement request is not found, no action is taken and false is returned.	description
cool:IdT id, cool:VersionT version	arguments
boolean successful	return
0.1	since ver.

updateMovementRequest	method
Updates a movement request in this NRP with data provided in the rtm (request to move) argument. Returns true if successful. If specified movement request is not found, no action is taken and false is returned. If successful the movement request version number is incremented.	description
cool:RequestToMoveT rtm	arguments

boolean successful	return
0.1	since ver.

addMovementRequest	method
Adds the movement request provided in the <code>rtm</code> (request to move) argument. Returns <code>true</code> if successful otherwise <code>false</code> .	description
<code>cool:RequestToMoveT rtm</code>	arguments
boolean successful	return
0.1	since ver.

changePriority	method
Changes the priority that movement request identified by <code>id</code> is treated with, affects all versions of the movement request. Returns <code>true</code> if successful otherwise <code>false</code> .	description
<code>cool:IdT id, string priority</code>	arguments
boolean successful	return
0.1	since ver.

getMovementRequestList	method
Retrieves a list of all the movement requests available in the NRP. Returns a <code>MoveRequestList</code> that has 0 or more movement requests. The <code>MoveRequestList</code> only contains a limited amount of information contained in a complete movement request.	description
none	arguments
<code>cool:MoveRequestList rtmList</code>	return
removed 0.3, replaced <code>getAllMovementRequests</code> 0.9	since ver.

8.2 Movement Request – Coalition Server

Description:

Provides access and management functionality for movement requests in the CIE. This Movement Request web service is deployed in the Coalition Server and references both Coalition Server and NRP data.

Dependencies: Movement Request – NRP instances

Web Service Style: Document

8.2.1 Interface Definition

getMovementRequestList	method
Retrieves a list of all movement requests available in the Coalition Server. Returns a <code>MoveRequestList</code> that has 0 or more movement requests.	description
none	arguments
<code>cool:MoveRequestList rtmList</code>	return
0.1	since ver.

getMovementRequestListFromAny	method
Retrieves a list of all movement requests available in the CIE (contains data from both the Coalition Server and NRP). Returns a <code>MoveRequestList</code> that has 0 or more movement requests. This web services exploits the <code>getMovementRequestList</code> interface on the coalition server and NRP.	description
none	arguments
<code>cool:MoveRequestList rtmList</code>	return
0.1	since ver.

getMovementRequest	method
Retrieves a movement request identified by the <code>id</code> and <code>ver</code> arguments from the CID. Returns a movement request or a fault message if not found.	description
<code>cool:IdT id, cool:VersionT version</code>	arguments
<code>cool:RequestToMoveT rtm</code>	return
0.3	since ver.

getMovementRequestFromAny	method
Retrieves a movement request identified by the <code>id</code> and <code>ver</code> arguments from the CID, or if not found in the CID, from any NRP. Returns a movement request or fault message if not found.	description
<code>cool:IdT id, cool:VersionT version</code>	arguments
<code>cool:RequestToMoveT rtm</code>	return
0.3	since ver.

removeMovementRequest	method
Removes movement request identified by <code>id</code> and <code>ver</code> arguments from the CID. Returns the number of instances of the movement request removed successfully.	description
<code>cool:IdT id, cool:VersionT version</code>	arguments
<code>boolean successfulFlag</code>	return
0.3	since ver.

removeMovementRequestFromAny	method
Removes movement requests identified by the <code>id</code> and <code>ver</code> arguments from the CID and NRPs. Returns the number of instances of the movement request that were removed successfully.	description
<code>cool:IdT id, cool:VersionT version</code>	arguments
<code>long noRemoved</code>	return
0.3	since ver.

updateMovementRequest	method
Updates the movement request in CID. Returns <code>true</code> if successful. If an error occurs or the movement request to update is not found, a SOAP fault message is returned.	description

cool:RequestToMoveT rtm	arguments
Boolean successFlag	return
0.3	since ver.

updateMovementRequestFromAny	method
Updates the movement request in the CID, or if not found, in NRP(s). Returns the number of successful updates completed.	description
cool:RequestToMoveT rtm	arguments
long noUpdated	return
0.3	since ver.

changePriority	method
Changes the priority of movement request identified by id . This operation affects all movement request versions. Returns true if successful otherwise false is returned.	description
cool:IdT id, string priority	arguments
boolean successful	return
0.1	since ver.

changePriorityFromAny	method
Changes the priority of movement request identified by id in CID, or if not found, in NRP(s). Returns number of successful priority changes made.	description
cool:IdT id, string priority	arguments
long noChanged	return
0.3	since ver.

getMovementRequestByOrg	method
Retrieves movement requests that belong to the organization identified by org . Returns a MoveRequestList that has 0 or more movement requests.	description
cool:OrgInfoT org	arguments
cool:MoveRequestList rtmList	return
0.1	since ver.

getMovementRequestByOrgFromAny	method
Retrieves movement requests that belong to the organization identified by org . Returns a MoveRequestList that has 0 or more movement requests.	description
cool:OrgInfoT org	arguments
cool:MoveRequestList rtmList	return
0.5	since ver.

addMovementRequest	method
Adds movement requests to the CID. Returns true if successful, otherwise a SOAP fault message is returned.	description
cool:MoveRequestList rtmList	arguments
Boolean successFlag	return
0.1	since ver.

8.3 Coalition Movement Requirement Plan – Coalition Server

Description:

Provides access and management functionality for Coalition Movement Requirement Plans (CMRP). Movement plans contain information from multiple nations and therefore only reside on the coalition server; therefore, services that operate on this data only appear on the coalition server.

Dependencies: none

Web Service Style: Document

8.3.1 Interface Definition

getMovementPlanList	method
Retrieves all coalition movement requirement plans currently in the CID. Only a small subset of data is returned, not the complete movement plan.	description
none	arguments
<code>cool:MovementPlanList cmrpList</code>	return
0.1	since ver.

getMovementPlan	method
Retrieves the movement plan identified by <code>planId</code> and <code>version</code> . Returns a <code>MovementPlanT</code> or SOAP fault message if the movement plan can't be found.	description
<code>cool:IdT planId, string version</code>	arguments
<code>cool:MovementPlanT plan</code>	return
0.1	since ver.

deleteMovementPlan	method
Removes a movement plan identified by <code>planId</code> and <code>version</code> from the CID. Returns <code>true</code> if successful <code>false</code> otherwise.	description
<code>cool:IdT planId, cool:VersionT version</code>	arguments
<code>boolean</code>	return
removed 0.3 (replaced with <code>removeMovementPlan</code>)	since ver.

removeMovementPlan	method
Removes a movement plan identified by <code>planId</code> and <code>ver</code> from CID. Returns <code>true</code> if successful <code>false</code> otherwise.	description
<code>cool:IdT planId, cool:VersionT version</code>	arguments
<code>boolean</code>	return
0.3	since ver.

updateMovementPlan	method
Updates a movement plan in CID. Returns <code>true</code> if successful <code>false</code> otherwise.	description
<code>cool:MovementPlanT movPlan</code>	arguments
<code>boolean</code>	return
0.1	since ver.

addMovementPlan	method
Adds a new movement plan to the CID. Returns true if successful false otherwise.	description
cool:MovementPlanT movPlan	arguments
boolean	return
0.1	since ver.

addMovementRequirement	method
Adds a movement requirement to the CMRP specified. Returns true if successful false otherwise.	description
cool:IdT id, cool:VersionT version, cool:IdT sectionId, cool:MoveReqT movRequirement	arguments
boolean	return
0.3	since ver.

getMovementPlanSection	method
Gets the specified movement plan section.	description
cool:IdT id, cool:VersionT version, cool:IdT sectionId	arguments
cool:MovePlanSectT movSection	return
0.5	since ver.

updateMovementPlanSection	method
Updates the movement plan section in CID. Returns true if successful false otherwise.	description
cool:IdT id, cool:VersionT version, cool:IdT sectionId, cool:MovePlanSectT section	arguments
boolean	return
0.5	since ver.

addMovmenetPlanSection	method
Adds a movement plan section to the CMRP specified. Returns true if successful false otherwise.	description
cool:IdT id, cool:VersionT version, cool: MovePlanSectT section	arguments
boolean	return
0.5	since ver.

removeMovementPlanSection	Method
Removes the identified MovementPlanSection from CID. Returns true if successful, false otherwise.	Description
cool:IdT sectionId, cool:VersionT version, cool:IdT sectionId	Arguments
boolean	Return
0.5	since ver.

8.4 Organization – NRP

Description:

Provides access and management functionality for organizations that exist within an NRP. This is not a required service.

Dependencies: none

Web Service Style: Document

8.4.1 Interface Definition

getOrganization	Method
Retrieves the organization identified by orgId from the NRP.	Description
cool:IdT orgId	arguments
cool:OrgInfoT	return
0.1	since ver.

getOrganizationStructure	method
Retrieves the organization structure identified by orgId from the NRP.	description
cool:IdT orgId	arguments
cool:OrgStructT	return
0.1	since ver.

addOrganization	method
Adds a new organization to the NRP. Returns true if successful false otherwise.	description
cool:OrgInfoT org	arguments
boolean	return
0.1	since ver.

updateOrganization	method
Updates an organization in the NRP. Returns true if successful false otherwise.	description
cool:OrgInfoT org	arguments
boolean	return
0.1	since ver.

removeOrganization	method
Removes an organization from the NRP. Returns true if successful false otherwise.	description
cool:IdT orgId	arguments
boolean	return
0.1	since ver.

8.5 Organization – Coalition Server

Description:

Provides access and management functionality for organizations in the CIE. It is deployed on the coalition server and accesses both CID and NRP data.

Dependencies: Organization – NRP web services

Web Service Style: Document

8.5.1 Interface Definition

getOrganization	Method
Retrieves an organization identified by orgId from the CID.	Description
cool:IdT orgId	arguments
cool:OrgInfoT	return
0.9	since ver.

getOrganizationFromAny	method
Retrieves an organization identified by orgId from CID, or if not found, from the NRPs.	description
cool:IdT orgId	arguments
cool:OrgInfoT	return
0.1	since ver.

getOrganizationStructure	method
Retrieves an organization's structure identified by orgId from the CID.	description
cool:IdT orgId	arguments
cool:OrgStructT	return
0.9	since ver.

getOrganizationStructureFromAny	method
Retrieves an organization's structure identified by orgId from the CID and NRPs. Data returned is from the authoritative source (please note that the system currently can not easily determine the authoritative source of data).	description
cool:IdT orgId	arguments
cool:OrgStructT	return
0.1	since ver.

addOrganization	method
Adds a new organization to the CID. Returns true if successful false otherwise.	description
cool:OrgInfoT org	arguments
boolean	return
0.1	since ver.

updateOrganization	method
Updates an organization in the CID. Returns true if successful false otherwise.	description
cool:OrgInfoT org	arguments

boolean	return
0.9	since ver.

updateOrganizationFromAny	method
Updates an organization in CID, or if not found, in an NRP. Returns true if successful false otherwise.	description
cool:OrgInfoT	arguments
boolean	return
0.1	since ver.

removeOrganization	method
Removes an organization from the CID. Returns true if successful false otherwise.	description
cool:IdT orgId	arguments
boolean	return
0.9	since ver.

removeOrganizationFromAny	method
Removes an organization identified by orgId from the CID, or if not found, from an NRP. Returns the number of organization instances removed.	description
cool:IdT orgId	arguments
long noOrgs	return
0.1	since ver.

getOrganizationIdByCountryCode	method
Retrieves organization Ids for organizations belonging to the country code specified.	description
cool:LocationT/Code countryCode	arguments
cool:IdTList	return
0.1	since ver.

8.6 Personnel - NRP

Description:

Provides access and management functionality for personnel data in an NRP.

Dependencies: none

Web Service Style: Document

8.6.1 Interface Definition

getPerson	method
Retrieves a person identified by personId from an NRP.	description
cool:IdT personId	arguments
cool:PersonT	return
0.1	since ver.

getPeopleInOrg	method
Retrieves people belonging to an organization identified by	description

orgId in this NRP.	
cool:IdT orgId	arguments
cool:PersonTList	return
0.1	since ver.

updatePerson	method
Updates person data for the provided person in this NRP.	description
cool:PersonT person	arguments
boolean	return
0.1	since ver.

removePerson	method
Removes the person identified by personId from this NRP.	description
cool:IdT personId	arguments
boolean	return
0.1	since ver.

addPerson	method
Adds a person to this NRP.	description
cool:PersonT person	arguments
boolean	return
0.1	since ver.

8.7 Personnel – Coalition Server

Description:

Provides access and management functionality for personnel data in the CID.

Dependencies: Personnel – NRP web services

Web Service Style: Document

8.7.1 Interface Definition

getPerson	method
Retrieves a person identified by personId from the CID.	description
cool:IdT personId	arguments
cool:PersonT	return
0.9	since ver.

getPersonFromAny	method
Retrieves a person identified by personId from CID, or if not found, from an NRP.	description
cool:IdT personId	arguments
cool:PerstonT	return
0.1	since ver.

getPeopleInOrg	method
Retrieves people belonging to the organization identified by orgId in the CID.	description
cool:IdT orgId	arguments

cool:PersonTList	return
0.9	since ver.

getPeopleInOrgFromAny	method
Retrieves people belonging to the organization identified by orgId from CID, or if not found, from an NRP.	description
cool:IdT orgId	arguments
cool:PersonTList	return
0.1	since ver.

updatePerson	method
Updates person data for the provided person in the CID.	description
cool:PersonT person	arguments
Boolean	return
0.9	since ver.

updatePersonFromAny	method
Updates person in the CID, or if not found, in the NRPs.	description
cool:PersonT person	arguments
Boolean	return
0.1	since ver.

removePerson	method
Removes the person identified by personId from the CID.	description
cool:IdT personId	arguments
Boolean	Return
0.9	since ver.

removePersonFromAny	Method
Removes person identified by personId from CID, or if not found, from NRPs.	Description
cool:IdT personId	Arguments
Boolean	Return
0.1	since ver.

addPerson	method
Adds a person to the CID.	description
cool:PersonT person	arguments
Boolean	return
0.9	since ver.

8.8 Equipment – NRP

Description:

Manages equipment data in movement requests that is stored in an NRP.

Dependencies: none

Web Service Style: Document

8.8.1 Interface Definition

getEquipment	method
Retrieves equipment identified by equipId from the NRP.	description
cool:IdT equipId	arguments
cool:BaseEquipT item	return
0.1	since ver.

addEquipment	method
Adds this equipment to the NRP.	description
cool:BaseEquipT equipment	arguments
boolean	return
0.1	since ver.

updateEquipment	method
Updates this equipment in the NRP.	description
cool:BaseEquipT equipment	arguments
boolean	return
0.1	since ver.

removeEquipment	method
Removes equipment identified by equipId from an NRP.	description
cool:IdT equipId	arguments
boolean	return
0.1	since ver.

getEquipmentFromOrganization	method
Retrieve all equipment associated with the organization identified by orgId from an NRP.	description
cool:IdT orgId	arguments
cool:BaseEquipTList	return
0.1	since ver.

8.9 Equipment – Coalition Server

Description:

Manages equipment data in movement requests that is stored in the Coalition Server and NRP databases.

Dependencies: Equipment – NRP web services

Web Service Style: Document

8.9.1 Interface Definition

getEquipment	method
Retrieves equipment identified by equipId in the CID, or if not found, in the NRP.	description
cool:IdT equipId	arguments
cool:BaseEquipT item	return
0.1	since ver.

getEquipmentFromAny	method
Retrieves equipment identified by <code>equipId</code> from CID, or if not found, in the NRP.	description
<code>cool:IdT equipId</code>	arguments
<code>cool:BaseEquipT</code>	return
0.1	since ver.

updateEquipment	method
Updates this equipment in the CID.	description
<code>cool:BaseEquipT equipment</code>	arguments
<code>boolean</code>	return
0.9	since ver.

updateEquipmentFromAny	method
Updates this equipment in the CID, or if not found, in NRP.	description
<code>cool:BaseEquipT equipment</code>	arguments
<code>boolean</code>	return
0.1	since ver.

removeEquipment	method
Removes equipment identified by <code>equipId</code> from the CID.	description
<code>cool:IdT equipId</code>	arguments
<code>boolean</code>	return
0.9	since ver.

removeEquipmentFromAny	method
Removes equipment identified by <code>equipId</code> from CID, or if not found, in an NRP.	description
<code>cool:IdT equipId</code>	arguments
<code>boolean</code>	return
0.1	since ver.

getEquipmentFromOrganization	method
Retrieve all equipment associated with the organization identified by <code>orgId</code> from CID.	description
<code>cool:IdT orgId</code>	arguments
<code>cool:BaseEquipTList</code>	return
0.9	since ver.

getEquipmentFromOrgFromAny	method
Retrieve all equipment associated with the organization identified by <code>orgId</code> from CID, or if not found, in an NRP.	description
<code>cool:IdT orgId</code>	arguments
<code>cool:BaseEquipTList</code>	return
0.1	since ver.

addEquipment	method
Adds this equipment to the CID.	description
<code>cool:BaseEquipT equipment</code>	arguments

boolean	return
0.9	since ver.

8.10 Cargo In-Transit Visibility (ITV)

Description:

Manages In-Transit Visibility (ITV) of cargo. The same interface is used for both NRP and Coalition Server instances.

Dependencies: none

Web Service Style: Document

8.10.1 Interface Definition

addITVRecords	method
Adds Cargo ITV records.	description
cool: CargoITVList	arguments
long noUpdated	return
0.1, 0.9 changed ITVRecordList type to CargoITVList	since ver.

getITVbyId	method
Retrieves Cargo ITV records for the specified coalition equipment Id.	description
cool: IdT	arguments
cool: CargoITVList	return
0.1, 0.9 changed ITVRecordList type to CargoITVList	since ver.

getITVbyNSN	method
Retrieves Cargo ITV records for the specified NATO Stock Number (NSN).	description
string NSN	arguments
cool: CargoITVList	return
0.1, 0.9 changed ITVRecordList type to CargoITVList	since ver.

getITVbyPOD	method
Retrieves Cargo ITV records for the specified Point of Disembarkation (POD).	description
string POD	arguments
cool: CargoITVList	return
0.1, 0.9 changed ITVRecordList type to CargoITVList	since ver.

getITVbyPOE	method
Retrieves Cargo ITV records for the specified Point of Embarkation (POE).	description
string POE	arguments
cool: CargoITVList	return
0.1, 0.9 changed ITVRecordList type to CargoITVList	since ver.

getITVbyNomenclature	method
Retrieves Cargo ITV records for the specified	description

nomenclature.	
string nomenclature	arguments
cool: CargoITVList	return
0.1, 0.9 changed ITVRecordList type to CargoITVList	since ver.

getITVbyOrigin	method
Retrieves Cargo ITV records for the specified origin.	description
cool:LocationT origin	arguments
cool: CargoITVList	return
0.1, 0.9 changed ITVRecordList type to CargoITVList	since ver.

getITVbyDestination	method
Retrieves Cargo ITV records for the specified destination.	description
cool:LocationT destination	arguments
cool: CargoITVList	return
0.1, 0.9 changed ITVRecordList type to CargoITVList	since ver.

getITVbySupplyClass	method
Retrieves Cargo ITV records for the specified supplyClass.	description
string supplyClass	arguments
cool: CargoITVList	return
0.1, 0.9 changed ITVRecordList type to CargoITVList	since ver.

getITVbyPerson	method
Retrieves Cargo ITV records for the specified person who requested the item being tracked.	description
cool:PersonT person	arguments
cool: CargoITVList	return
0.1, 0.9 changed ITVRecordList type to CargoITVList	since ver.

getITVbyOrg	method
Retrieves Cargo ITV records for the specified organizationId.	description
cool: IdT organizationId.	arguments
cool: CargoITVList	return
0.1, 0.9 changed ITVRecordList type to CargoITVList	since ver.

updateITVRecords	method
Updates Cargo ITV records.	description
cool: CargoITVList	arguments
long noUpdated	return
0.1, 0.9 changed ITVRecordList type to CargoITVList	since ver.

removeITVRecords	method
Removes Cargo ITV records.	description
cool:IdTList cargoIds	arguments
long noRemoved	return
0.1, 0.9 changed argument type	since ver.

8.11 Coalition ID – Coalition Server

Description:

Creates and keeps track of IDs being used in the CIE. The return value is the start of an ID number sequence. It is up to the user to generate actual IDs.

Dependencies: none

Web Service Style: RPC

8.11.1 Interface Definition

getList	method
Retrieves the next available <code>id</code> sequence start. ID sequences are 100 IDs long and clients are required to generate the following IDs in the sequence.	description
<code>string idType</code>	arguments
<code>cool:IdT startOfIdList</code>	return
0.1	since ver.

lookupId	method
Retrieves the ID type for a <code>coalitionId</code> .	description
<code>cool:IdT coalitionId</code>	arguments
<code>string idType</code>	return
0.1	since ver.

8.12 Language – Coalition Server

Description:

This service allows for content of documents to be translated between languages.

Dependencies: none

Web Service Style: Document

8.12.1 Interface Definition

convertDocument	method
Converts text values in the XML source document from the source language to the target language. If a source or target language dictionary does not exist then a SOAP fault message is returned.	description
<code>xml:anyType document, string sourceLang, string targetLang</code>	arguments
<code>xml:anyType document</code>	return
0.1	since ver.

8.13 Unit Conversion – Coalition Server

Description:

Converts units of measure. Example use could be converting kilograms into pounds.

Dependencies: none

Web Service Style: RPC

8.13.1 Interface Definition

convertUnit	method
Converts <code>value</code> from source unit system to target unit system. Eg. feet to metres.	description
<code>string value, string sourceUnits, string targetUnits</code>	arguments
<code>string newValue</code>	return
0.1	since ver.

8.14 Location – Coalition Server

Description:

This is a utility service to resolve locations that use various data representations.

Dependencies: none

Web Service Style: Document

8.14.1 Interface Definition

lookupUNLOC	method
Performs a United National Location (UNLOC) code lookup for the given location.	description
<code>cool:LocationT loc</code>	arguments
<code>string UNLOC</code>	return
0.1	since ver.

lookupICAO	method
Performs an International Civil Aviation Organisation (ICAO) code lookup for the given location.	description
<code>cool:LocationT loc</code>	arguments
<code>string ICAO</code>	return
0.1	since ver.

lookupPortCode	method
Performs a Port code lookup for the given location.	description
<code>cool:LocationT loc</code>	arguments
<code>string portCode</code>	return
0.1	since ver.

convertToUTM	method
Converts a location to Universal Transverse Mercator (UTM) representation.	description
<code>cool:LocationT loc</code>	arguments
<code>string UTMLocation</code>	return
0.1	since ver.

convertToLatLon	method
Converts a location to its latitude and longitude representation.	description
<code>cool:LocationT loc</code>	arguments

string lat, string lon	return
0.1	since ver.

getLocationInfo	method
Retrieves full location details for a location identified by locId.	description
cool:IdT locId	arguments
cool:LocationT	return
0.1	since ver.

9 Future Directions

The web services and definitions described are only the initial definitions for CTL ACTD. Over the coming years CTL ACTD will grow and support additional functionality. The areas currently lacking in the CTL web service are:

- Authoritative Data sources
- Coalition IDs guidelines
- Updated XML Schema definition
- UDDI registries guidelines
- Web services security
- Emerging web service standards

Each issue is described in the following sections.

9.1 Authoritative Data Sources

Further work and design is required to create a true authoritative definition and searching scheme. Currently it is almost impossible to find out if a data item is the authoritative source or where to find the true authoritative source. Adding extra metadata to the XML Schema or creation of a data item registry are two methods that may help solve this issue.

9.2 Coalition IDs

The current implementation and use of XML data item IDs is very loose and hence creates problems when using these. Tracking the use of IDs so that data can be updated with the same ID is the single biggest problem, especially when transforming from national to coalition data formats. Better guidelines for creating, using and managing IDs is therefore required.

9.3 Update XML Schema Definition

In the current XML Schema definition there is a large number of superfluous entities not required in the coalition. The extra items are a legacy from a few US developed applications and are no longer required. By redeveloping the schema, new data items or constructs can be added to support features such as authoritative sources, extensibility, and inclusion of NATO (or other coalition nations) data models. The key requirement is to allow the schema to be extensible so that new items can be defined over time or to store auxiliary data. Support for auxiliary data would allow nations to embed national specific data in the coalition data that can be used by nations that understand these extensions but will not break coalition applications and services.

9.4 UDDI Registry Guidelines

Currently the UDDI registry is used in a very ad-hoc manner creating confusion between national developers and creating interoperability issues. The biggest problem is in how web services are defined and registered. Guidelines and specifications therefore need to be created that define exactly how to register services, what these services are named in the registry, and how the registered services are to be used. These guidelines are key to the correct operation of the

Information Manager web services that forward web service messages out to the NRP servers.

9.5 Web Services Security

Although this document does briefly discuss the use of web services security, at the time of design and development of CTL ACTD web services the security standards were still developing, hence no detailed implementation was created. Future work will require CTL services to include security services as defined in the WS-I Basic Security Profiles [13].

9.6 Emerging Web Service Standards

The web service environment is continually evolving with new standards being developed, merged, or replaced. The core standards (SOAP, WSDL, and UDDI) are reasonably stable with only minor revisions made since CTL ACTD. The WSDL standard has recently gone through a major revision that results in a standard for defining services that is easier to understand and it is flexible. It is recommended that future versions of CTL utilise the WSDL 2.0 specification [17]. Other emerging standards that may be applicable and should be explored further are:

- WS-Discovery – Dynamically discover web services without the need of a UDDI registry.
- WS-Addressing – Standard method of addressing services and items on a network.
- BPEL4WS – Business Process definition and execution language.
- OWL-S – Ontology language to help describe services.

There are other standards in addition to those identified above. The emerging web service space is always in a state of flux so identifying these here would be pointless as they will have changed by the time of future development. Before conducting further CTL development new standards should be identified and reviewed as to their applicability for use in CTL.

Appendix A: Sample WSDLs for CTL ACTD Web Services

This appendix contains samples of all WSDL files used to implement the web service interfaces described in Section 8 on page 22. Table 1 below shows a list of sample WSDL files and the page numbers they can be found on. Electronic copies of all WSDL files are contained on the CD-ROM attached to this report. If no CD-ROM is present please contact Egon Kuster from Command and Control Division in the Defence Science and Technology Organisation.

Table 1 – List of Sample WSDL files

1 - MovementRequestNRP.....	46
2 - MovementRequestCS.....	49
3 - MovementRequestServiceSOAP.....	53
4 - CMRP.....	54
5 - CMRPServiceSOAP.....	57
6 - OrganizationNRP.....	59
7 - OrganizationCS.....	61
8 - OrganizationServiceSOAP.....	64
9 - PersonnelNRP.....	65
10 - PersonnelCS.....	67
11 - PersonnelServiceSOAP.....	70
12 - EquipmentNRP.....	71
13 - EquipmentCS.....	73
14 - EquipmentServiceSOAP.....	76
15 - ITV.....	77
16 - ITVServiceSOAP.....	81
17 - CoalitionID.....	82
18 - Language.....	83
19 - Unit.....	84
20 - Location.....	85
21 - LocationServiceSOAP.....	87

1 - MovementRequestNRP

```

<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:y="http://ws.actd.ctl"
  xmlns:cool="http://coalition/xml/schema" targetNamespace="http://ws.actd.ctl">
  <types>
    <xs:schema targetNamespace="http://coalition/xml/schema">
      <xs:import namespace="http://coalition/xml/schema"
        schemaLocation="./coalitionSchema/MovementRequestServiceSOAP.xsd"/>
    </xs:schema>
  </types>
  <message name="getMovementRequestListRequest">
    <part name="request" element="cool:getMovementRequestListRequest"/>
  </message>
  <message name="getMovementRequestListResponse">
    <part name="response" element="cool:MoveRequestList"/>
  </message>
  <message name="changePriorityRequest">
    <part name="request" element="cool:changePriorityRequest"/>
  </message>
  <message name="changePriorityResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="addMovementRequestRequest">
    <part name="request" element="cool:addMovementRequestRequest"/>
  </message>
  <message name="addMovementRequestResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="removeMovementRequestRequest">
    <part name="request" element="cool:removeMovementRequestRequest"/>
  </message>
  <message name="removeMovementRequestResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="getMovementRequestRequest">
    <part name="request" element="cool:getMovementRequestRequest"/>
  </message>
  <message name="getMovementRequestResponse">
    <part name="response" element="cool:RequestToMove"/>
  </message>
  <message name="validateMovementRequestRequest">
    <part name="request" element="cool:validateMovementRequestRequest"/>
  </message>
  <message name="validateMovementRequestResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="updateMovementRequestRequest">
    <part name="request" element="cool:updateMovementRequestRequest"/>
  </message>
  <message name="updateMovementRequestResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <portType name="MovementRequestPort">
    <operation name="changePriority">
      <input message="y:changePriorityRequest"/>
      <output message="y:changePriorityResponse"/>
    </operation>
    <operation name="validateMovementRequest">
      <input message="y:validateMovementRequestRequest"/>
      <output message="y:validateMovementRequestResponse"/>
    </operation>
    <operation name="addMovementRequest">
      <input message="y:addMovementRequestRequest"/>
      <output message="y:addMovementRequestResponse"/>
    </operation>
    <operation name="updateMovementRequest">
      <input message="y:updateMovementRequestRequest"/>
      <output message="y:updateMovementRequestResponse"/>
    </operation>
    <operation name="removeMovementRequest">
      <input message="y:removeMovementRequestRequest"/>
      <output message="y:removeMovementRequestResponse"/>
    </operation>
  </portType>

```

```

</operation>
<operation name="getMovementRequest">
  <input message="y:getMovementRequestRequest"/>
  <output message="y:getMovementRequestResponse"/>
</operation>
<operation name="getMovementRequestList">
  <input message="y:getMovementRequestListRequest"/>
  <output message="y:getMovementRequestListResponse"/>
</operation>
</portType>
<binding name="MovementRequestBinding" type="y:MovementRequestPort">
  <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="changePriority">
    <soap:operation soapAction="changePriority" style="document"/>
    <input>
      <soap:body parts="request" use="literal"/>
    </input>
    <output>
      <soap:body parts="response" use="literal"/>
    </output>
  </operation>
  <operation name="validateMovementRequest">
    <soap:operation soapAction="validateMovementRequest" style="document"/>
    <input>
      <soap:body parts="request" use="literal"/>
    </input>
    <output>
      <soap:body parts="response" use="literal"/>
    </output>
  </operation>
  <operation name="addMovementRequest">
    <soap:operation soapAction="addMovementRequest" style="document"/>
    <input>
      <soap:body parts="request" use="literal"/>
    </input>
    <output>
      <soap:body parts="response" use="literal"/>
    </output>
  </operation>
  <operation name="updateMovementRequest">
    <soap:operation soapAction="updateMovementRequest" style="document"/>
    <input>
      <soap:body parts="request" use="literal"/>
    </input>
    <output>
      <soap:body parts="response" use="literal"/>
    </output>
  </operation>
  <operation name="removeMovementRequest">
    <soap:operation soapAction="removeMovementRequest" style="document"/>
    <input>
      <soap:body parts="request" use="literal"/>
    </input>
    <output>
      <soap:body parts="response" use="literal"/>
    </output>
  </operation>
  <operation name="getMovementRequest">
    <soap:operation soapAction="getMovementRequest" style="document"/>
    <input>
      <soap:body parts="request" use="literal"/>
    </input>
    <output>
      <soap:body parts="response" use="literal"/>
    </output>
  </operation>
  <operation name="getMovementRequestList">
    <soap:operation soapAction="getAllMovementRequests" style="document"/>
    <input>
      <soap:body parts="request" use="literal"/>
    </input>
    <output>
      <soap:body parts="response" use="literal"/>
    </output>
  </operation>
</binding>

```

```
<service name="MovementRequest">  
  <port name="MovementRequest" binding="y:MovementRequestBinding">  
    <soap:address location="http://sampleLocation.com"/>  
  </port>  
</service>  
</definitions>
```

2 – MovementRequestCS

```

<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:y="http://ws.actd.ctl"
  xmlns:cool="http://coalition/xml/schema" targetNamespace="http://ws.actd.ctl">
  <types>
    <xs:schema targetNamespace="http://coalition/xml/schema">
      <xs:import namespace="http://coalition/xml/schema"
        schemaLocation="./coalitionSchema/MovementRequestServiceSOAP.xsd"/>
      </xs:schema>
    </types>
    <message name="getMovementRequestListRequest">
      <part name="request" element="cool:getMovementRequestListRequest"/>
    </message>
    <message name="getMovementRequestListResponse">
      <part name="response" element="cool:MoveRequestList"/>
    </message>
    <message name="changePriorityRequest">
      <part name="request" element="cool:changePriorityRequest"/>
    </message>
    <message name="changePriorityResponse">
      <part name="response" element="cool:booleanResponse"/>
    </message>
    <message name="addMovementRequestRequest">
      <part name="request" element="cool:addMovementRequestRequest"/>
    </message>
    <message name="addMovementRequestResponse">
      <part name="response" element="cool:booleanResponse"/>
    </message>
    <message name="removeMovementRequestRequest">
      <part name="request" element="cool:removeMovementRequestRequest"/>
    </message>
    <message name="removeMovementRequestResponse">
      <part name="response" element="cool:booleanResponse"/>
    </message>
    <message name="getMovementRequestRequest">
      <part name="request" element="cool:getMovementRequestRequest"/>
    </message>
    <message name="getMovementRequestResponse">
      <part name="response" element="cool:RequestToMove"/>
    </message>
    <message name="validateMovementRequestRequest">
      <part name="request" element="cool:validateMovementRequestRequest"/>
    </message>
    <message name="validateMovementRequestResponse">
      <part name="response" element="cool:booleanResponse"/>
    </message>
    <message name="updateMovementRequestRequest">
      <part name="request" element="cool:updateMovementRequestRequest"/>
    </message>
    <message name="updateMovementRequestResponse">
      <part name="response" element="cool:booleanResponse"/>
    </message>
    <message name="getMovementRequestListFromAnyRequest">
      <part name="request" element="cool:getMovementRequestListFromAnyRequest"/>
    </message>
    <message name="getMovementRequestListFromAnyResponse">
      <part name="response" element="cool:MoveRequestList"/>
    </message>
    <message name="getMovementRequestFromAnyRequest">
      <part name="request" element="cool:getMovementRequestFromAnyRequest"/>
    </message>
    <message name="getMovementRequestFromAnyResponse">
      <part name="response" element="cool:RequestToMove"/>
    </message>
    <message name="removeMovementRequestFromAny">
      <part name="request" element="cool:removeMovementRequestFromAnyRequest"/>
    </message>
    <message name="removeMovementRequestFromAnyResponse">
      <part name="response" element="cool:longResponse"/>
    </message>
    <message name="updateMovementRequestFromAnyRequest">
      <part name="request" element="cool:updateMovementRequestFromAnyRequest"/>
    </message>

```

```

</message>
<message name="updateMovementRequestFromAnyResponse">
  <part name="response" element="cool:longResponse"/>
</message>
<message name="changePriorityFromAnyRequest">
  <part name="request" element="cool:changePriorityFromAnyRequest"/>
</message>
<message name="changePriorityFromAnyResponse">
  <part name="response" element="cool:longResponse"/>
</message>
<message name="getMovementRequestByOrgRequest">
  <part name="request" element="cool:getMovementRequestByOrgRequest"/>
</message>
<message name="getMovementRequestByOrgResponse">
  <part name="response" element="cool:MoveRequestList"/>
</message>
<message name="getMovementRequestByOrgFromAnyRequest">
  <part name="request" element="cool:getMovementRequestByOrgFromAnyRequest"/>
</message>
<message name="getMovementRequestByOrgFromAnyResponse">
  <part name="response" element="cool:MoveRequestList"/>
</message>
<portType name="MovementRequestPort">
  <operation name="changePriority">
    <input message="y:changePriorityRequest"/>
    <output message="y:changePriorityResponse"/>
  </operation>
  <operation name="changePriorityFromAny">
    <input message="y:changePriorityFromAnyRequest"/>
    <output message="y:changePriorityFromAnyResponse"/>
  </operation>
  <operation name="validateMovementRequest">
    <input message="y:validateMovementRequestRequest"/>
    <output message="y:validateMovementRequestResponse"/>
  </operation>
  <operation name="addMovementRequest">
    <input message="y:addMovementRequestRequest"/>
    <output message="y:addMovementRequestResponse"/>
  </operation>
  <operation name="updateMovementRequest">
    <input message="y:updateMovementRequestRequest"/>
    <output message="y:updateMovementRequestResponse"/>
  </operation>
  <operation name="updateMovementRequestFromAny">
    <input message="y:updateMovementRequestFromAnyRequest"/>
    <output message="y:updateMovementRequestFromAnyResponse"/>
  </operation>
  <operation name="removeMovementRequest">
    <input message="y:removeMovementRequestRequest"/>
    <output message="y:removeMovementRequestResponse"/>
  </operation>
  <operation name="removeMovementRequestFromAny">
    <input message="y:removeMovementRequestFromAny"/>
    <output message="y:removeMovementRequestFromAnyResponse"/>
  </operation>
  <operation name="getMovementRequest">
    <input message="y:getMovementRequestRequest"/>
    <output message="y:getMovementRequestResponse"/>
  </operation>
  <operation name="getMovementRequestFromAny">
    <input message="y:getMovementRequestFromAnyRequest"/>
    <output message="y:getMovementRequestFromAnyResponse"/>
  </operation>
  <operation name="getMovementRequestList">
    <input message="y:getMovementRequestListRequest"/>
    <output message="y:getMovementRequestListResponse"/>
  </operation>
  <operation name="getMovementRequestListFromAny">
    <input message="y:getMovementRequestListFromAnyRequest"/>
    <output message="y:getMovementRequestListFromAnyResponse"/>
  </operation>
  <operation name="getMovementRequestByOrg">
    <input message="y:getMovementRequestByOrgRequest"/>
    <output message="y:getMovementRequestByOrgResponse"/>
  </operation>
  <operation name="getMovementRequestByOrgFromAny">

```

```

        <input message="y:getMovementRequestByOrgFromAnyRequest"/>
        <output message="y:getMovementRequestByOrgFromAnyResponse"/>
    </operation>
</portType>
<binding name="MovementRequestBinding" type="y:MovementRequestPort">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="validateMovementRequest">
        <soap:operation soapAction="validateMovementRequest" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
    <operation name="changePriority">
        <soap:operation soapAction="changePriority" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
    <operation name="changePriorityFromAny">
        <soap:operation soapAction="changePriorityFromAny" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
    <operation name="addMovementRequest">
        <soap:operation soapAction="addMovementRequest" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
    <operation name="updateMovementRequest">
        <soap:operation soapAction="updateMovementRequest" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
    <operation name="updateMovementRequestFromAny">
        <soap:operation soapAction="updateMovementRequestFromAny" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
    <operation name="removeMovementRequest">
        <soap:operation soapAction="removeMovementRequest" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
    <operation name="removeMovementRequestFromAny">
        <soap:operation soapAction="removeMovementRequestFromAny"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>

```

```

        </output>
    </operation>
    <operation name="getMovementRequest">
        <soap:operation soapAction="getMovementRequest" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
    <operation name="getMovementRequestFromAny">
        <soap:operation soapAction="getMovementRequestFromAny" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
    <operation name="getMovementRequestList">
        <soap:operation soapAction="getAllMovementRequests" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
    <operation name="getMovementRequestListFromAny">
        <soap:operation soapAction="getMovementRequestListFromAny" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
    <operation name="getMovementRequestByOrg">
        <soap:operation soapAction="getMovementRequestByOrg" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
    <operation name="getMovementRequestByOrgFromAny">
        <soap:operation soapAction="getMovementRequestByOrgFromAny"
style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
</binding>
<service name="MovementRequest">
    <port name="MovementRequest" binding="y:MovementRequestBinding">
        <soap:address location="http://sampleLocation.com"/>
    </port>
</service>
</definitions>

```

3 – MovementRequestServiceSOAP

```

<xs:schema targetNamespace="http://coalition/xml/schema"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:cool="http://coalition/xml/schema">
  <xs:include schemaLocation="./common/BaseTypes.xsd"/>
  <xs:include schemaLocation="./MovRequest.xsd"/>
  <xs:element name="booleanResponse" type="xs:boolean">
    <xs:annotation>
      <xs:documentation>A boolean response.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="longResponse" type="xs:long">
    <xs:annotation>
      <xs:documentation>A long response.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="changePriorityRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="requestId" type="cool:IdT"/>
        <xs:element name="priority" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="changePriorityFromAnyRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="requestId" type="cool:IdT"/>
        <xs:element name="priority" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="addMovementRequestRequest" type="cool:RequestToMoveT"/>
  <xs:element name="getMovementRequestListRequest"/>
  <xs:element name="getMovementRequestListFromAnyRequest"/>
  <xs:element name="getMovementRequestRequest">
    <xs:complexType>
      <xs:attribute name="requestId" type="cool:IdT" use="required"/>
      <xs:attribute name="requestVersion" type="cool:VersionT" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="getMovementRequestFromAnyRequest">
    <xs:complexType>
      <xs:attribute name="requestId" type="cool:IdT" use="required"/>
      <xs:attribute name="requestVersion" type="cool:VersionT" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="getMovementRequestByOrgRequest" type="cool:OrgInfoT"/>
  <xs:element name="getMovementRequestByOrgFromAnyRequest" type="cool:OrgInfoT"/>
  <xs:element name="updateMovementRequestRequest" type="cool:RequestToMoveT"/>
  <xs:element name="updateMovementRequestFromAnyRequest" type="cool:RequestToMoveT"/>
  <xs:element name="removeMovementRequestRequest">
    <xs:complexType>
      <xs:attribute name="requestId" type="cool:IdT" use="required"/>
      <xs:attribute name="requestVersion" type="cool:VersionT" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="removeMovementRequestFromAnyRequest">
    <xs:complexType>
      <xs:attribute name="requestId" type="cool:IdT" use="required"/>
      <xs:attribute name="requestVersion" type="cool:VersionT" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="validateMovementRequestRequest" type="cool:RequestToMoveT"/>
</xs:schema>

```

4 - CMRP

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:y="http://ws.actd.ctl"
  xmlns:cool="http://coalition/xml/schema" targetNamespace="http://ws.actd.ctl">
  <types>
    <xs:schema targetNamespace="http://coalition/xml/schema">
      <xs:import namespace="http://coalition/xml/schema"
        schemaLocation="./coalitionSchema/CMRPSERVICESOAP.xsd"/>
    </xs:schema>
  </types>
  <message name="getMovementPlanListRequest">
    <part name="request" element="cool:getMovementPlanListRequest"/>
  </message>
  <message name="getMovementPlanListResponse">
    <part name="response" element="cool:MovementPlanList"/>
  </message>
  <message name="getMovementPlanRequest">
    <part name="request" element="cool:getMovementPlanRequest"/>
  </message>
  <message name="getMovementPlanResponse">
    <part name="response" element="cool:MovementPlan"/>
  </message>
  <message name="removeMovementPlanRequest">
    <part name="request" element="cool:removeMovementPlanRequest"/>
  </message>
  <message name="removeMovementPlanResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="updateMovementPlanRequest">
    <part name="request" element="cool:updateMovementPlanRequest"/>
  </message>
  <message name="updateMovementPlanResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="addMovementPlanRequest">
    <part name="request" element="cool:addMovementPlanRequest"/>
  </message>
  <message name="addMovementPlanResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="addMovementRequirementRequest">
    <part name="request" element="cool:addMovementRequirementRequest"/>
  </message>
  <message name="addMovementRequirementResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="getMovementPlanSectionRequest">
    <part name="request" element="cool:getMovementPlanSectionRequest"/>
  </message>
  <message name="getMovementPlanSectionResponse">
    <part name="response" element="cool:MovePlanSect"/>
  </message>
  <message name="updateMovementPlanSectionRequest">
    <part name="request" element="cool:updateMovementPlanRequest"/>
  </message>
  <message name="updateMovementPlanSectionResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="addMovementPlanSectionRequest">
    <part name="request" element="cool:addMovementPlanSectionRequest"/>
  </message>
  <message name="addMovementPlanSectionResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="removeMovementPlanSectionRequest">
    <part name="request" element="cool:removeMovementPlanSectionRequest"/>
  </message>
  <message name="removeMovementPlanSectionResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <portType name="CMRPPort">

```

```

<operation name="getMovementPlanList">
  <input message="y:getMovementPlanListRequest"/>
  <output message="y:getMovementPlanListResponse"/>
</operation>
<operation name="getMovementPlan">
  <input message="y:getMovementPlanRequest"/>
  <output message="y:getMovementPlanResponse"/>
</operation>
<operation name="removeMovementPlan">
  <input message="y:removeMovementPlanRequest"/>
  <output message="y:removeMovementPlanResponse"/>
</operation>
<operation name="updateMovementPlan">
  <input message="y:updateMovementPlanRequest"/>
  <output message="y:updateMovementPlanResponse"/>
</operation>
<operation name="addMovementPlan">
  <input message="y:addMovementPlanRequest"/>
  <output message="y:addMovementPlanResponse"/>
</operation>
<operation name="addMovementRequirement">
  <input message="y:addMovementRequirementRequest"/>
  <output message="y:addMovementRequirementResponse"/>
</operation>
<operation name="getMovementPlanSection">
  <input message="y:getMovementPlanSectionRequest"/>
  <output message="y:getMovementPlanSectionResponse"/>
</operation>
<operation name="updateMovementPlanSection">
  <input message="y:updateMovementPlanSectionRequest"/>
  <output message="y:updateMovementPlanSectionResponse"/>
</operation>
<operation name="addMovementPlanSection">
  <input message="y:addMovementPlanSectionRequest"/>
  <output message="y:addMovementPlanSectionResponse"/>
</operation>
<operation name="removeMovementPlanSection">
  <input message="y:removeMovementPlanSectionRequest"/>
  <output message="y:removeMovementPlanSectionResponse"/>
</operation>
</portType>
<binding name="CMRPBidning" type="y:CMRPPort">
  <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="getMovementPlanList">
    <soap:operation soapAction="getMovementPlanList" style="document"/>
    <input>
      <soap:body parts="request" use="literal"/>
    </input>
    <output>
      <soap:body parts="response" use="literal"/>
    </output>
  </operation>
  <operation name="getMovementPlan">
    <soap:operation soapAction="getMovementPlan" style="document"/>
    <input>
      <soap:body parts="request" use="literal"/>
    </input>
    <output>
      <soap:body parts="response" use="literal"/>
    </output>
  </operation>
  <operation name="removeMovementPlan">
    <soap:operation soapAction="removeMovementPlan" style="document"/>
    <input>
      <soap:body parts="request" use="literal"/>
    </input>
    <output>
      <soap:body parts="response" use="literal"/>
    </output>
  </operation>
  <operation name="updateMovementPlan">
    <soap:operation soapAction="updateMovementPlan" style="document"/>
    <input>
      <soap:body parts="request" use="literal"/>
    </input>
    <output>

```

```

        <soap:body parts="response" use="literal"/>
    </output>
</operation>
<operation name="addMovementPlan">
    <soap:operation soapAction="addMovementPlan" style="document"/>
    <input>
        <soap:body parts="request" use="literal"/>
    </input>
    <output>
        <soap:body parts="response" use="literal"/>
    </output>
</operation>
<operation name="addMovementRequirement">
    <soap:operation soapAction="addMovementRequirement" style="document"/>
    <input>
        <soap:body parts="request" use="literal"/>
    </input>
    <output>
        <soap:body parts="response" use="literal"/>
    </output>
</operation>
<operation name="getMovementPlanSection">
    <soap:operation soapAction="getMovementPlanSection" style="document"/>
    <input>
        <soap:body parts="request" use="literal"/>
    </input>
    <output>
        <soap:body parts="response" use="literal"/>
    </output>
</operation>
<operation name="updateMovementPlanSection">
    <soap:operation soapAction="updateMovementPlanSection" style="document"/>
    <input>
        <soap:body parts="request" use="literal"/>
    </input>
    <output>
        <soap:body parts="response" use="literal"/>
    </output>
</operation>
<operation name="addMovementPlanSection">
    <soap:operation soapAction="urn:#addMovementPlanSection"/>
    <input>
        <soap:body use="literal"/>
    </input>
    <output>
        <soap:body use="literal"/>
    </output>
</operation>
<operation name="removeMovementPlanSection">
    <soap:operation soapAction="urn:#removeMovementPlanSection"/>
    <input>
        <soap:body use="literal"/>
    </input>
    <output>
        <soap:body use="literal"/>
    </output>
</operation>
</binding>
<service name="CMRP">
    <port name="CMRP" binding="y:CMRPBidding">
        <soap:address location="http://sampleLocation.com"/>
    </port>
</service>
</definitions>

```

5 – CMRPServiceSOAP

```

<xs:schema targetNamespace="http://coalition/xml/schema"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:cool="http://coalition/xml/schema">
  <xs:include schemaLocation=".MovPlan.xsd"/>
  <xs:element name="booleanResponse" type="xs:boolean">
    <xs:annotation>
      <xs:documentation>A boolean response.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="longResponse" type="xs:long">
    <xs:annotation>
      <xs:documentation>A long response.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="getMovementPlanListRequest" type="xs:string"/>
  <xs:element name="getMovementPlanRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="planId" type="cool:IdT"/>
        <xs:element name="version" type="cool:VersionT"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="removeMovementPlanRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="planId" type="cool:IdT"/>
        <xs:element name="version" type="cool:VersionT"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="updateMovementPlanRequest" type="cool:MovementPlanT"/>
  <xs:element name="addMovementPlanRequest" type="cool:MovementPlanT"/>
  <xs:element name="addMovementRequirementRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="planId" type="cool:IdT"/>
        <xs:element name="version" type="cool:VersionT"/>
        <xs:element name="sectionId" type="cool:IdT"/>
        <xs:element name="moveRequirement" type="cool:MoveReqT"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="getMovementPlanSectionRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="planId" type="cool:IdT"/>
        <xs:element name="version" type="cool:VersionT"/>
        <xs:element name="sectionId" type="cool:IdT"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="updateMovementPlanSectionRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="planId" type="cool:IdT"/>
        <xs:element name="version" type="cool:VersionT"/>
        <xs:element name="sectionId" type="cool:IdT"/>
        <xs:element name="moveRequirement" type="cool:MovePlanSectT"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="addMovementPlanSectionRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="planId" type="cool:IdT"/>
        <xs:element name="version" type="cool:VersionT"/>
        <xs:element name="moveRequirement" type="cool:MovePlanSectT"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="removeMovementPlanSectionRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="planId" type="cool:IdT"/>
        <xs:element name="version" type="cool:VersionT"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

```
        <xs:element name="sectionId" type="cool:IdT"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="MovePlanSect" type="cool:MovePlanSectT"/>
</xs:schema>
```

6 – OrganizationNRP

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:cool="http://coalition/xml/schema" xmlns:y="http://ws.actd.ctl"
  targetNamespace="http://ws.actd.ctl">
  <types>
    <xs:schema targetNamespace="http://coalition/xml/schema">
      <xs:import namespace="http://coalition/xml/schema"
        schemaLocation="./coalitionSchema/OrganizationServiceSOAP.xsd"/>
    </xs:schema>
  </types>
  <message name="getOrganizationRequest">
    <part name="request" element="cool:getOrganizationRequest"/>
  </message>
  <message name="getOrganizationResponse">
    <part name="response" element="cool:OrgInfo"/>
  </message>
  <message name="getOrganizationStructureRequest">
    <part name="request" element="cool:getOrganizationStructureRequest"/>
  </message>
  <message name="getOrganizationStructureResponse">
    <part name="response" element="cool:OrgStruct"/>
  </message>
  <message name="addOrganizationRequest">
    <part name="request" element="cool:addOrganizationRequest"/>
  </message>
  <message name="addOrganizationResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="updateOrganizationRequest">
    <part name="request" element="cool:updateOrganizationRequest"/>
  </message>
  <message name="updateOrganizationResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="removeOrganizationRequest">
    <part name="request" element="cool:removeOrganizationRequest"/>
  </message>
  <message name="removeOrganizationResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <portType name="OrganizationPort">
    <operation name="getOrganization">
      <input message="y:getOrganizationRequest"/>
      <output message="y:getOrganizationResponse"/>
    </operation>
    <operation name="getOrganizationStructure">
      <input message="y:getOrganizationStructureRequest"/>
      <output message="y:getOrganizationStructureResponse"/>
    </operation>
    <operation name="addOrganization">
      <input message="y:addOrganizationRequest"/>
      <output message="y:addOrganizationResponse"/>
    </operation>
    <operation name="updateOrganization">
      <input message="y:updateOrganizationRequest"/>
      <output message="y:updateOrganizationResponse"/>
    </operation>
    <operation name="removeOrganization">
      <input message="y:removeOrganizationRequest"/>
      <output message="y:removeOrganizationResponse"/>
    </operation>
  </portType>
  <binding name="OrganizationBinding" type="y:OrganizationPort">
    <soap:binding style="document"
      transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="getOrganization">
      <soap:operation soapAction="getOrganization" style="document"/>
      <input>
        <soap:body parts="request" use="literal"/>
      </input>
    </operation>
  </binding>
</definitions>

```

```

        <output>
          <soap:body parts="response" use="literal"/>
        </output>
      </operation>
      <operation name="getOrganizationStructure">
        <soap:operation soapAction="getOrganizationStructure" style="document"/>
        <input>
          <soap:body parts="request" use="literal"/>
        </input>
        <output>
          <soap:body parts="response" use="literal"/>
        </output>
      </operation>
      <operation name="addOrganization">
        <soap:operation soapAction="addOrganization" style="document"/>
        <input>
          <soap:body parts="request" use="literal"/>
        </input>
        <output>
          <soap:body parts="response" use="literal"/>
        </output>
      </operation>
      <operation name="updateOrganization">
        <soap:operation soapAction="updateOrganization" style="document"/>
        <input>
          <soap:body parts="request" use="literal"/>
        </input>
        <output>
          <soap:body parts="response" use="literal"/>
        </output>
      </operation>
      <operation name="removeOrganization">
        <soap:operation soapAction="removeOrganization" style="document"/>
        <input>
          <soap:body parts="request" use="literal"/>
        </input>
        <output>
          <soap:body parts="response" use="literal"/>
        </output>
      </operation>
    </binding>
    <service name="Organization">
      <port name="Organization" binding="y:OrganizationBinding">
        <soap:address location="http://sampleLocation.com"/>
      </port>
    </service>
  </definitions>

```

7 – OrganizationCS

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:cool="http://coalition/xml/schema" xmlns:y="http://ws.actd.ctl"
  targetNamespace="http://ws.actd.ctl">
  <types>
    <xs:schema targetNamespace="http://coalition/xml/schema">
      <xs:import namespace="http://coalition/xml/schema"
        schemaLocation="./coalitionSchema/OrganizationServiceSOAP.xsd"/>
    </xs:schema>
  </types>
  <message name="getOrganizationRequest">
    <part name="request" element="cool:getOrganizationRequest"/>
  </message>
  <message name="getOrganizationResponse">
    <part name="response" element="cool:OrgInfo"/>
  </message>
  <message name="getOrganizationFromAnyRequest">
    <part name="request" element="cool:getOrganizationFromAnyRequest"/>
  </message>
  <message name="getOrganizationFromAnyResponse">
    <part name="response" element="cool:OrgInfo"/>
  </message>
  <message name="getOrganizationStructureRequest">
    <part name="request" element="cool:getOrganizationStructureRequest"/>
  </message>
  <message name="getOrganizationStructureResponse">
    <part name="response" element="cool:OrgStruct"/>
  </message>
  <message name="getOrganizationStructureFromAnyRequest">
    <part name="request" element="cool:getOrganizationStructureFromAnyRequest"/>
  </message>
  <message name="getOrganizationStructureFromAnyResponse">
    <part name="response" element="cool:OrgStruct"/>
  </message>
  <message name="addOrganizationRequest">
    <part name="request" element="cool:addOrganizationRequest"/>
  </message>
  <message name="addOrganizationResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="updateOrganizationRequest">
    <part name="request" element="cool:updateOrganizationRequest"/>
  </message>
  <message name="updateOrganizationResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="updateOrganizationFromAnyRequest">
    <part name="request" element="cool:updateOrganizationFromAnyRequest"/>
  </message>
  <message name="updateOrganizationFromAnyResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="removeOrganizationRequest">
    <part name="request" element="cool:removeOrganizationRequest"/>
  </message>
  <message name="removeOrganizationResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="removeOrganizationFromAnyRequest">
    <part name="request" element="cool:removeOrganizationFromAnyRequest"/>
  </message>
  <message name="removeOrganizationFromAnyResponse">
    <part name="response" element="cool:longResponse"/>
  </message>
  <message name="getOrganizationIdByCountryCodeRequest">
    <part name="request" element="cool:getOrganizationIdByCountryCodeRequest"/>
  </message>
  <message name="getOrganizationIdByCountryCodeResponse">
    <part name="response" element="cool:IdList"/>
  </message>
```

```

<portType name="OrganizationPort">
  <operation name="getOrganization">
    <input message="y:getOrganizationRequest"/>
    <output message="y:getOrganizationResponse"/>
  </operation>
  <operation name="getOrganizationFromAny">
    <input message="y:getOrganizationFromAnyRequest"/>
    <output message="y:getOrganizationFromAnyResponse"/>
  </operation>
  <operation name="getOrganizationStructure">
    <input message="y:getOrganizationStructureRequest"/>
    <output message="y:getOrganizationStructureResponse"/>
  </operation>
  <operation name="getOrganizationStructureFromAny">
    <input message="y:getOrganizationStructureFromAnyRequest"/>
    <output message="y:getOrganizationStructureFromAnyResponse"/>
  </operation>
  <operation name="addOrganization">
    <input message="y:addOrganizationRequest"/>
    <output message="y:addOrganizationResponse"/>
  </operation>
  <operation name="updateOrganization">
    <input message="y:updateOrganizationRequest"/>
    <output message="y:updateOrganizationResponse"/>
  </operation>
  <operation name="updateOrganizationFromAny">
    <input message="y:updateOrganizationFromAnyRequest"/>
    <output message="y:updateOrganizationFromAnyResponse"/>
  </operation>
  <operation name="removeOrganization">
    <input message="y:removeOrganizationRequest"/>
    <output message="y:removeOrganizationResponse"/>
  </operation>
  <operation name="removeOrganizationFromAny">
    <input message="y:removeOrganizationFromAnyRequest"/>
    <output message="y:removeOrganizationFromAnyResponse"/>
  </operation>
  <operation name="getOrganizationIdByCountryCode">
    <input message="y:getOrganizationIdByCountryCodeRequest"/>
    <output message="y:getOrganizationIdByCountryCodeResponse"/>
  </operation>
</portType>
<binding name="OrganizationBinding" type="y:OrganizationPort">
  <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="getOrganization">
    <soap:operation soapAction="getOrganization" style="document"/>
    <input>
      <soap:body parts="request" use="literal"/>
    </input>
    <output>
      <soap:body parts="response" use="literal"/>
    </output>
  </operation>
  <operation name="getOrganizationFromAny">
    <soap:operation soapAction="getOrganizationFromAny" style="document"/>
    <input>
      <soap:body parts="request" use="literal"/>
    </input>
    <output>
      <soap:body parts="response" use="literal"/>
    </output>
  </operation>
  <operation name="getOrganizationStructure">
    <soap:operation soapAction="getOrganizationStructure" style="document"/>
    <input>
      <soap:body parts="request" use="literal"/>
    </input>
    <output>
      <soap:body parts="response" use="literal"/>
    </output>
  </operation>
  <operation name="getOrganizationStructureFromAny">
    <soap:operation soapAction="getOrganizationStructureFromAny"
style="document"/>
    <input>

```

```

        <soap:body parts="request" use="literal"/>
    </input>
    <output>
        <soap:body parts="response" use="literal"/>
    </output>
</operation>
<operation name="addOrganization">
    <soap:operation soapAction="addOrganization" style="document"/>
    <input>
        <soap:body parts="request" use="literal"/>
    </input>
    <output>
        <soap:body parts="response" use="literal"/>
    </output>
</operation>
<operation name="updateOrganization">
    <soap:operation soapAction="updateOrganization" style="document"/>
    <input>
        <soap:body parts="request" use="literal"/>
    </input>
    <output>
        <soap:body parts="response" use="literal"/>
    </output>
</operation>
<operation name="updateOrganizationFromAny">
    <soap:operation soapAction="updateOrganizationFromAny" style="document"/>
    <input>
        <soap:body parts="request" use="literal"/>
    </input>
    <output>
        <soap:body parts="response" use="literal"/>
    </output>
</operation>
<operation name="removeOrganization">
    <soap:operation soapAction="removeOrganization" style="document"/>
    <input>
        <soap:body parts="request" use="literal"/>
    </input>
    <output>
        <soap:body parts="response" use="literal"/>
    </output>
</operation>
<operation name="removeOrganizationFromAny">
    <soap:operation soapAction="removeOrganizationFromAny" style="document"/>
    <input>
        <soap:body parts="request" use="literal"/>
    </input>
    <output>
        <soap:body parts="response" use="literal"/>
    </output>
</operation>
<operation name="getOrganizationIdByCountryCode">
    <soap:operation soapAction="getOrganizationIdByCountryCode"
style="document"/>
    <input>
        <soap:body parts="request" use="literal"/>
    </input>
    <output>
        <soap:body parts="response" use="literal"/>
    </output>
</operation>
</binding>
<service name="Organization">
    <port name="Organization" binding="y:OrganizationBinding">
        <soap:address location="http://sampleLocation.com"/>
    </port>
</service>
</definitions>

```

8 – OrganizationServiceSOAP

```

<xs:schema targetNamespace="http://coalition/xml/schema"
xmlns:cool="http://coalition/xml/schema" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation="./common/BaseTypes.xsd"/>
  <xs:element name="booleanResponse" type="xs:boolean">
    <xs:annotation>
      <xs:documentation>A boolean response.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="longResponse" type="xs:long">
    <xs:annotation>
      <xs:documentation>A long response.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="getOrganizationRequest" type="cool:IdT"/>
  <xs:element name="getOrganizationFromAnyRequest" type="cool:IdT"/>
  <xs:element name="getOrganizationStructureRequest" type="cool:IdT"/>
  <xs:element name="getOrganizationStructureFromAnyRequest" type="cool:IdT"/>
  <xs:element name="addOrganizationRequest" type="cool:OrgInfoT"/>
  <xs:element name="updateOrganizationRequest" type="cool:OrgInfoT"/>
  <xs:element name="updateOrganizationFromAnyRequest" type="cool:OrgInfoT"/>
  <xs:element name="removeOrganizationRequest" type="cool:IdT"/>
  <xs:element name="removeOrganizationFromAnyRequest" type="cool:IdT"/>
  <xs:element name="getOrganizationIdByCountryCodeRequest" type="cool:LocationT"/>
  <xs:element name="OrgInfo" type="cool:OrgInfoT"/>
  <xs:element name="OrgStruct" type="cool:OrgStructT"/>
  <xs:element name="IdList">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Id" type="cool:IdT"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

9 – PersonnelNRP

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:y="http://ws.actd.ctl"
  xmlns:cool="http://coalition/xml/schema" targetNamespace="http://ws.actd.ctl">
  <types>
    <xs:schema targetNamespace="http://coalition/xml/schema">
      <xs:import namespace="http://coalition/xml/schema"
        schemaLocation="../coalitionSchema/PersonnelServiceSOAP.xsd"/>
    </xs:schema>
  </types>
  <message name="getPeronsRequest">
    <part name="request" element="cool:getPersonRequest"/>
  </message>
  <message name="getPersonResponse">
    <part name="response" element="cool:Person"/>
  </message>
  <message name="getPeopleInOrgRequest">
    <part name="request" element="cool:getPeopleInOrgRequest"/>
  </message>
  <message name="getPeopleInOrgResponse">
    <part name="response" element="cool:PersonList"/>
  </message>
  <message name="updatePersonRequest">
    <part name="request" element="cool:updatePersonRequest"/>
  </message>
  <message name="updatePersonResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="removePersonRequest">
    <part name="request" element="cool:removePersonRequest"/>
  </message>
  <message name="removePersonResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="addPersonRequest">
    <part name="request" element="cool:addPersonRequest"/>
  </message>
  <message name="addPersonResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <portType name="PersonnelPort">
    <operation name="getPerson">
      <input message="y:getPeronsRequest"/>
      <output message="y:getPersonResponse"/>
    </operation>
    <operation name="getPeopleInOrg">
      <input message="y:getPeopleInOrgRequest"/>
      <output message="y:getPeopleInOrgResponse"/>
    </operation>
    <operation name="updatePerson">
      <input message="y:updatePersonRequest"/>
      <output message="y:updatePersonResponse"/>
    </operation>
    <operation name="removePerson">
      <input message="y:removePersonRequest"/>
      <output message="y:removePersonResponse"/>
    </operation>
    <operation name="addPerson">
      <input message="y:addPersonRequest"/>
      <output message="y:addPersonResponse"/>
    </operation>
  </portType>
  <binding name="PersonnelBinding" type="y:PersonnelPort">
    <soap:binding style="document"
      transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="getPerson">
      <soap:operation soapAction="getPerson" style="document"/>
      <input>
        <soap:body parts="request" use="literal"/>
      </input>
      <output>

```

```

        <soap:body parts="response" use="literal"/>
    </output>
</operation>
<operation name="getPeopleInOrg">
    <soap:operation soapAction="getPeopleInOrg" style="document"/>
    <input>
        <soap:body parts="request" use="literal"/>
    </input>
    <output>
        <soap:body parts="response" use="literal"/>
    </output>
</operation>
<operation name="updatePerson">
    <soap:operation soapAction="updatePerson" style="document"/>
    <input>
        <soap:body parts="request" use="literal"/>
    </input>
    <output>
        <soap:body parts="response" use="literal"/>
    </output>
</operation>
<operation name="removePerson">
    <soap:operation soapAction="removePerson" style="document"/>
    <input>
        <soap:body parts="request" use="literal"/>
    </input>
    <output>
        <soap:body parts="response" use="literal"/>
    </output>
</operation>
<operation name="addPerson">
    <soap:operation soapAction="addPerson" style="document"/>
    <input>
        <soap:body parts="request" use="literal"/>
    </input>
    <output>
        <soap:body parts="response" use="literal"/>
    </output>
</operation>
</binding>
<service name="Personnel">
    <port name="Personnel" binding="y:PersonnelBinding">
        <soap:address location="http://sampleLocation.com"/>
    </port>
</service>
</definitions>

```

10 – PersonnelCS

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:y="http://ws.actd.ctl"
  xmlns:cool="http://coalition/xml/schema" targetNamespace="http://ws.actd.ctl">
  <types>
    <xs:schema targetNamespace="http://coalition/xml/schema">
      <xs:import namespace="http://coalition/xml/schema"
        schemaLocation="../coalitionSchema/PersonnelServiceSOAP.xsd"/>
    </xs:schema>
  </types>
  <message name="getPeronsRequest">
    <part name="request" element="cool:getPersonRequest"/>
  </message>
  <message name="getPersonResponse">
    <part name="response" element="cool:Person"/>
  </message>
  <message name="getPersonFromAnyRequest">
    <part name="request" element="cool:getPersonFromAnyRequest"/>
  </message>
  <message name="getPersonFromAnyResponse">
    <part name="response" element="cool:Person"/>
  </message>
  <message name="getPeopleInOrgRequest">
    <part name="request" element="cool:getPeopleInOrgRequest"/>
  </message>
  <message name="getPeopleInOrgResponse">
    <part name="response" element="cool:PersonList"/>
  </message>
  <message name="getPeopleInOrgFromAnyRequest">
    <part name="request" element="cool:getPeopleInOrgFromAnyRequest"/>
  </message>
  <message name="getPeopleInOrgFromAnyResponse">
    <part name="response" element="cool:PersonList"/>
  </message>
  <message name="updatePersonRequest">
    <part name="request" element="cool:updatePersonRequest"/>
  </message>
  <message name="updatePersonResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="updatePersonFromAnyRequest">
    <part name="request" element="cool:updatePersonFromAnyRequest"/>
  </message>
  <message name="updatePersonFromAnyResponse">
    <part name="parameter" element="cool:booleanResponse"/>
  </message>
  <message name="removePersonRequest">
    <part name="request" element="cool:removePersonRequest"/>
  </message>
  <message name="removePersonResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="removePersonFromAnyRequest">
    <part name="request" element="cool:removePersonFromAnyRequest"/>
  </message>
  <message name="removePersonFromAnyResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="addPersonRequest">
    <part name="request" element="cool:addPersonRequest"/>
  </message>
  <message name="addPersonResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <portType name="PersonnelPort">
    <operation name="getPerson">
      <input message="y:getPeronsRequest"/>
      <output message="y:getPersonResponse"/>
    </operation>
    <operation name="getPersonFromAny">
      <input message="y:getPersonFromAnyRequest"/>

```

```

        <output message="y:getPersonFromAnyResponse"/>
    </operation>
    <operation name="getPeopleInOrg">
        <input message="y:getPeopleInOrgRequest"/>
        <output message="y:getPeopleInOrgResponse"/>
    </operation>
    <operation name="getPeopleInOrgFromAny">
        <input message="y:getPeopleInOrgFromAnyRequest"/>
        <output message="y:getPeopleInOrgFromAnyResponse"/>
    </operation>
    <operation name="updatePerson">
        <input message="y:updatePersonRequest"/>
        <output message="y:updatePersonResponse"/>
    </operation>
    <operation name="updatePersonFromAny">
        <input message="y:updatePersonFromAnyRequest"/>
        <output message="y:updatePersonFromAnyResponse"/>
    </operation>
    <operation name="removePerson">
        <input message="y:removePersonRequest"/>
        <output message="y:removePersonResponse"/>
    </operation>
    <operation name="removePersonFromAny">
        <input message="y:removePersonFromAnyRequest"/>
        <output message="y:removePersonFromAnyResponse"/>
    </operation>
    <operation name="addPerson">
        <input message="y:addPersonRequest"/>
        <output message="y:addPersonResponse"/>
    </operation>
</portType>
<binding name="PersonnelBinding" type="y:PersonnelPort">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="getPerson">
        <soap:operation soapAction="getPerson" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
    <operation name="getPersonFromAny">
        <soap:operation soapAction="getPersonFromAny" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
    <operation name="getPeopleInOrg">
        <soap:operation soapAction="getPeopleInOrg" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
    <operation name="getPeopleInOrgFromAny">
        <soap:operation soapAction="getPeopleInOrgFromAny" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
    <operation name="updatePerson">
        <soap:operation soapAction="updatePerson" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>

```

```

        <soap:body parts="response" use="literal"/>
      </output>
    </operation>
    <operation name="updatePersonFromAny">
      <soap:operation soapAction="updatePersonFromAny" style="document"/>
      <input>
        <soap:body parts="request" use="literal"/>
      </input>
      <output>
        <soap:body parts="response" use="literal"/>
      </output>
    </operation>
    <operation name="removePerson">
      <soap:operation soapAction="removePerson" style="document"/>
      <input>
        <soap:body parts="request" use="literal"/>
      </input>
      <output>
        <soap:body parts="response" use="literal"/>
      </output>
    </operation>
    <operation name="removePersonFromAny">
      <soap:operation soapAction="removePersonFromAny" style="document"/>
      <input>
        <soap:body parts="request" use="literal"/>
      </input>
      <output>
        <soap:body parts="response" use="literal"/>
      </output>
    </operation>
    <operation name="addPerson">
      <soap:operation soapAction="addPerson" style="document"/>
      <input>
        <soap:body parts="request" use="literal"/>
      </input>
      <output>
        <soap:body parts="response" use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="Personnel">
    <port name="Personnel" binding="y:PersonnelBinding">
      <soap:address location="http://sampleLocation.com"/>
    </port>
  </service>
</definitions>

```

11 – PersonnelServiceSOAP

```

<xs:schema targetNamespace="http://coalition/xml/schema"
xmlns:cool="http://coalition/xml/schema" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation="./common/BaseTypes.xsd"/>
  <xs:include schemaLocation="./Person.xsd"/>
  <xs:include schemaLocation="./PersonList.xsd"/>
  <xs:element name="booleanResponse" type="xs:boolean">
    <xs:annotation>
      <xs:documentation>A boolean response.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="longResponse" type="xs:long">
    <xs:annotation>
      <xs:documentation>A long response.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="getPersonRequest" type="cool:IdT"/>
  <xs:element name="getPersonFromAnyRequest" type="cool:IdT"/>
  <xs:element name="getPeopleInOrgRequest" type="cool:IdT"/>
  <xs:element name="getPeopleInOrgFromAnyRequest" type="cool:IdT"/>
  <xs:element name="updatePersonRequest" type="cool:PersonT"/>
  <xs:element name="updatePersonFromAnyRequest" type="cool:PersonT"/>
  <xs:element name="removePersonRequest" type="cool:IdT"/>
  <xs:element name="removePersonFromAnyRequest" type="cool:IdT"/>
  <xs:element name="addPersonRequest" type="cool:PersonT"/>
</xs:schema>

```

12 – EquipmentNRP

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:y="http://ws.actd.ctl"
  xmlns:cool="http://coalition/xml/schema" targetNamespace="http://ws.actd.ctl">
  <types>
    <xs:schema targetNamespace="http://coalition/xml/schema">
      <xs:import namespace="http://coalition/xml/schema"
        schemaLocation="./coalitionSchema/EquipmentServiceSOAP.xsd"/>
    </xs:schema>
  </types>
  <message name="getEquipmentRequest">
    <part name="request" element="cool:getEquipmentRequest"/>
  </message>
  <message name="getEquipmentResponse">
    <part name="response" element="cool:Item"/>
  </message>
  <message name="addEquipmentRequest">
    <part name="request" element="cool:addEquipmentRequest"/>
  </message>
  <message name="addEquipmentResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="updateEquipmentRequest">
    <part name="request" element="cool:updateEquipmentRequest"/>
  </message>
  <message name="updateEquipmentResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="removeEquipmentRequest">
    <part name="request" element="cool:removeEquipmentRequest"/>
  </message>
  <message name="removeEquipmentResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="getEquipmentFromOrganizationRequest">
    <part name="request" element="cool:getEquipmentFromOrganizationRequest"/>
  </message>
  <message name="getEquipmentFromOrganizationResponse">
    <part name="response" element="cool:EquipmentList"/>
  </message>
  <portType name="EquipmentPort">
    <operation name="getEquipment">
      <input message="y:getEquipmentRequest"/>
      <output message="y:getEquipmentResponse"/>
    </operation>
    <operation name="addEquipment">
      <input message="y:addEquipmentRequest"/>
      <output message="y:addEquipmentResponse"/>
    </operation>
    <operation name="updateEquipment">
      <input message="y:updateEquipmentRequest"/>
      <output message="y:updateEquipmentResponse"/>
    </operation>
    <operation name="removeEquipment">
      <input message="y:removeEquipmentRequest"/>
      <output message="y:removeEquipmentResponse"/>
    </operation>
    <operation name="getEquipmentFromOrganization">
      <input message="y:getEquipmentFromOrganizationRequest"/>
      <output message="y:getEquipmentFromOrganizationResponse"/>
    </operation>
  </portType>
  <binding name="EquipmentBinding" type="y:EquipmentPort">
    <soap:binding style="document"
      transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="getEquipment">
      <soap:operation soapAction="getEquipment" style="document"/>
      <input>
        <soap:body parts="request" use="literal"/>
      </input>
      <output>

```

```

        <soap:body parts="response" use="literal"/>
    </output>
</operation>
<operation name="addEquipment">
    <soap:operation soapAction="addEquipment" style="document"/>
    <input>
        <soap:body parts="request" use="literal"/>
    </input>
    <output>
        <soap:body parts="response" use="literal"/>
    </output>
</operation>
<operation name="updateEquipment">
    <soap:operation soapAction="updateEquipment" style="document"/>
    <input>
        <soap:body parts="request" use="literal"/>
    </input>
    <output>
        <soap:body parts="response" use="literal"/>
    </output>
</operation>
<operation name="removeEquipment">
    <soap:operation soapAction="removeEquipment" style="document"/>
    <input>
        <soap:body parts="request" use="literal"/>
    </input>
    <output>
        <soap:body parts="response" use="literal"/>
    </output>
</operation>
<operation name="getEquipmentFromOrganization">
    <soap:operation soapAction="getEquipmentFromOrganization" style="document"/>
    <input>
        <soap:body parts="request" use="literal"/>
    </input>
    <output>
        <soap:body parts="response" use="literal"/>
    </output>
</operation>
</binding>
<service name="Equipment">
    <port name="Equipment" binding="y:EquipmentBinding">
        <soap:address location="http://sampleLocation.com"/>
    </port>
</service>
</definitions>

```

13 – EquipmentCS

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:y="http://ws.actd.ctl"
  xmlns:cool="http://coalition/xml/schema" targetNamespace="http://ws.actd.ctl">
  <types>
    <xs:schema targetNamespace="http://coalition/xml/schema">
      <xs:import namespace="http://coalition/xml/schema"
        schemaLocation="./coalitionSchema/EquipmentServiceSOAP.xsd"/>
    </xs:schema>
  </types>
  <message name="getEquipmentRequest">
    <part name="request" element="cool:getEquipmentRequest"/>
  </message>
  <message name="getEquipmentResponse">
    <part name="response" element="cool:Item"/>
  </message>
  <message name="addEquipmentRequest">
    <part name="request" element="cool:addEquipmentRequest"/>
  </message>
  <message name="addEquipmentResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="updateEquipmentRequest">
    <part name="request" element="cool:updateEquipmentRequest"/>
  </message>
  <message name="updateEquipmentResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="removeEquipmentRequest">
    <part name="request" element="cool:removeEquipmentRequest"/>
  </message>
  <message name="removeEquipmentResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="getEquipmentFromOrganizationRequest">
    <part name="request" element="cool:getEquipmentFromOrganizationRequest"/>
  </message>
  <message name="getEquipmentFromOrganizationResponse">
    <part name="response" element="cool:EquipmentList"/>
  </message>
  <message name="getEquipmentFromAnyRequest">
    <part name="request" element="cool:getEquipmentFromAnyRequest"/>
  </message>
  <message name="getEquipmentFromAnyResponse">
    <part name="response" element="cool:Item"/>
  </message>
  <message name="updateEquipmentFromAnyRequest">
    <part name="request" element="cool:updateEquipmentFromAnyRequest"/>
  </message>
  <message name="updateEquipmentFromAnyResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="removeEquipmentFromAnyRequest">
    <part name="request" element="cool:removeEquipmentFromAnyRequest"/>
  </message>
  <message name="removeEquipmentFromAnyResponse">
    <part name="response" element="cool:booleanResponse"/>
  </message>
  <message name="getEquipmentFromOrganizationFromAnyRequest">
    <part name="request" element="cool:getEquipmentFromOrganizationFromAnyRequest"/>
  </message>
  <message name="getEquipmentFromOrganizationFromAnyResponse">
    <part name="response" element="cool:EquipmentList"/>
  </message>
  <portType name="EquipmentPort">
    <operation name="getEquipment">
      <input message="y:getEquipmentRequest"/>
      <output message="y:getEquipmentResponse"/>
    </operation>
    <operation name="getEquipmentFromAny">
      <input message="y:getEquipmentFromAnyRequest"/>

```

```

        <output message="y:getEquipmentFromAnyResponse"/>
    </operation>
    <operation name="addEquipment">
        <input message="y:addEquipmentRequest"/>
        <output message="y:addEquipmentResponse"/>
    </operation>
    <operation name="updateEquipment">
        <input message="y:updateEquipmentRequest"/>
        <output message="y:updateEquipmentResponse"/>
    </operation>
    <operation name="updateEquipmentFromAny">
        <input message="y:updateEquipmentFromAnyRequest"/>
        <output message="y:updateEquipmentFromAnyResponse"/>
    </operation>
    <operation name="removeEquipment">
        <input message="y:removeEquipmentRequest"/>
        <output message="y:removeEquipmentResponse"/>
    </operation>
    <operation name="removeEquipmentFromAny">
        <input message="y:removeEquipmentFromAnyRequest"/>
        <output message="y:removeEquipmentFromAnyResponse"/>
    </operation>
    <operation name="getEquipmentFromOrganization">
        <input message="y:getEquipmentFromOrganizationRequest"/>
        <output message="y:getEquipmentFromOrganizationResponse"/>
    </operation>
    <operation name="getEquipmentFromOrganizationFromAny">
        <input message="y:getEquipmentFromOrganizationFromAnyRequest"/>
        <output message="y:getEquipmentFromOrganizationFromAnyResponse"/>
    </operation>
</portType>
<binding name="EquipmentBinding" type="y:EquipmentPort">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="getEquipment">
        <soap:operation soapAction="getEquipment" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
    <operation name="getEquipmentFromAny">
        <soap:operation soapAction="getEquipmentFromAny" style="document"/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
    </operation>
    <operation name="addEquipment">
        <soap:operation soapAction="addEquipment" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
    <operation name="updateEquipment">
        <soap:operation soapAction="updateEquipment" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
    <operation name="updateEquipmentFromAny">
        <soap:operation soapAction="updateEquipmentFromAny" style="document"/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>

```

```

        <soap:body use="literal"/>
    </output>
</operation>
<operation name="removeEquipment">
    <soap:operation soapAction="removeEquipment" style="document"/>
    <input>
        <soap:body parts="request" use="literal"/>
    </input>
    <output>
        <soap:body parts="response" use="literal"/>
    </output>
</operation>
<operation name="removeEquipmentFromAny">
    <soap:operation soapAction="removeEquipmentFromAny" style="document"/>
    <input>
        <soap:body use="literal"/>
    </input>
    <output>
        <soap:body use="literal"/>
    </output>
</operation>
<operation name="getEquipmentFromOrganization">
    <soap:operation soapAction="getEquipmentFromOrganization" style="document"/>
    <input>
        <soap:body parts="request" use="literal"/>
    </input>
    <output>
        <soap:body parts="response" use="literal"/>
    </output>
</operation>
<operation name="getEquipmentFromOrganizationFromAny">
    <soap:operation soapAction="getEquipmentFromOrganizationFromAny"
style="document"/>
    <input>
        <soap:body use="literal"/>
    </input>
    <output>
        <soap:body use="literal"/>
    </output>
</operation>
</binding>
<service name="Equipment">
    <port name="Equipment" binding="y:EquipmentBinding">
        <soap:address location="http://sampleLocation.com"/>
    </port>
</service>
</definitions>

```

14 – EquipmentServiceSOAP

```

<xs:schema targetNamespace="http://coalition/xml/schema"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:cool="http://coalition/xml/schema">
  <xs:include schemaLocation="./common/BaseTypes.xsd"/>
  <xs:include schemaLocation="./EquipmentList.xsd"/>
  <xs:element name="booleanResponse" type="xs:boolean">
    <xs:annotation>
      <xs:documentation>A boolean response.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="getEquipmentRequest" type="cool:IdT"/>
  <xs:element name="getEquipmentFromAnyRequest" type="cool:IdT"/>
  <xs:element name="addEquipmentRequest" type="cool:BaseEquipT"/>
  <xs:element name="updateEquipmentRequest" type="cool:BaseEquipT"/>
  <xs:element name="updateEquipmentFromAnyRequest" type="cool:BaseEquipT"/>
  <xs:element name="removeEquipmentRequest" type="cool:IdT"/>
  <xs:element name="removeEquipmentFromAnyRequest" type="cool:IdT"/>
  <xs:element name="getEquipmentFromOrganizationRequest" type="cool:IdT"/>
  <xs:element name="getEquipmentFromOrganizationFromAnyRequest" type="cool:IdT"/>
  <xs:element name="Item" type="cool:BaseEquipT"/>
</xs:schema>

```

15 – ITV

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:y="http://ws.actd.ctl"
xmlns:cool="http://coalition/xml/schema" targetNamespace="http://ws.actd.ctl">
  <types>
    <xs:schema targetNamespace="http://coalition/xml/schema">
      <xs:import namespace="http://coalition/xml/schema"
schemaLocation="./coalitionSchema/ITVServiceSOAP.xsd"/>
    </xs:schema>
  </types>
  <message name="addITVRecordsRequest">
    <part name="request" element="cool:addITVRecordsRequest"/>
  </message>
  <message name="addITVRecordsResponse">
    <part name="response" element="cool:longResponse"/>
  </message>
  <message name="getITVByIdRequest">
    <part name="request" element="cool:getITVByIdRequest"/>
  </message>
  <message name="getITVByIdResponse">
    <part name="response" element="cool:CargoITVList"/>
  </message>
  <message name="getITVByNSNRequest">
    <part name="request" element="cool:getITVByNSNRequest"/>
  </message>
  <message name="getITVByNSNResponse">
    <part name="response" element="cool:CargoITVList"/>
  </message>
  <message name="getITVByPODRequest">
    <part name="request" element="cool:getITVByPODRequest"/>
  </message>
  <message name="getITVByPODResponse">
    <part name="response" element="cool:CargoITVList"/>
  </message>
  <message name="getITVByPOERequest">
    <part name="request" element="cool:getITVByPOERequest"/>
  </message>
  <message name="getITVByPOEResponse">
    <part name="response" element="cool:CargoITVList"/>
  </message>
  <message name="getITVByNomanc latureRequest">
    <part name="request" element="cool:getITVByNomanc latureRequest"/>
  </message>
  <message name="getITVByNomanc latureResponse">
    <part name="response" element="cool:CargoITVList"/>
  </message>
  <message name="getITVByOriginRequest">
    <part name="request" element="cool:getITVByOriginRequest"/>
  </message>
  <message name="getITVByOriginResponse">
    <part name="response" element="cool:CargoITVList"/>
  </message>
  <message name="getITVByDestinationRequest">
    <part name="request" element="cool:getITVByDestinationRequest"/>
  </message>
  <message name="getITVByDestinationResponse">
    <part name="response" element="cool:CargoITVList"/>
  </message>
  <message name="getITVBySupplyClassRequest">
    <part name="request" element="cool:getITVBySupplyClassRequest"/>
  </message>
  <message name="getITVBySupplyClassResponse">
    <part name="response" element="cool:CargoITVList"/>
  </message>
  <message name="getITVByPersonRequest">
    <part name="request" element="cool:getITVByPersonRequest"/>
  </message>
  <message name="getITVByPersonResponse">
    <part name="response" element="cool:CargoITVList"/>
  </message>
  <message name="getITVByOrgRequest">

```

```

    <part name="request" element="cool:getITVByOrgRequest"/>
</message>
<message name="getITVByOrgResponse">
  <part name="response" element="cool:CargoITVList"/>
</message>
<message name="updateITVRecordsRequest">
  <part name="request" element="cool:updateITVRecordsRequest"/>
</message>
<message name="updateITVRecordsResponse">
  <part name="response" element="cool:longResponse"/>
</message>
<message name="removeITVRecordsRequest">
  <part name="request" element="cool:removeITVRecordsRequest"/>
</message>
<message name="removeITVRecordsResponse">
  <part name="response" element="cool:longResponse"/>
</message>
<portType name="ITVPort">
  <operation name="addITVRecords">
    <input message="y:addITVRecordsRequest"/>
    <output message="y:addITVRecordsResponse"/>
  </operation>
  <operation name="getITVbyId">
    <input message="y:getITVByIdRequest"/>
    <output message="y:getITVByIdResponse"/>
  </operation>
  <operation name="getITVbyNSN">
    <input message="y:getITVByNSNRequest"/>
    <output message="y:getITVByNSNResponse"/>
  </operation>
  <operation name="getITVbyPOD">
    <input message="y:getITVByPODRequest"/>
    <output message="y:getITVByPODResponse"/>
  </operation>
  <operation name="getITVbyPOE">
    <input message="y:getITVByPOERequest"/>
    <output message="y:getITVByPOEResponse"/>
  </operation>
  <operation name="getITVbyNomanc lature">
    <input message="y:getITVByNomanc latureRequest"/>
    <output message="y:getITVByNomanc latureResponse"/>
  </operation>
  <operation name="getITVbyOrigin">
    <input message="y:getITVByOriginRequest"/>
    <output message="y:getITVByOriginResponse"/>
  </operation>
  <operation name="getITVbyDestination">
    <input message="y:getITVByDestinationRequest"/>
    <output message="y:getITVByDestinationResponse"/>
  </operation>
  <operation name="getITVbySupplyClass">
    <input message="y:getITVBySupplyClassRequest"/>
    <output message="y:getITVBySupplyClassResponse"/>
  </operation>
  <operation name="getITVbyPerson">
    <input message="y:getITVByPersonRequest"/>
    <output message="y:getITVByPersonResponse"/>
  </operation>
  <operation name="getITVbyOrg">
    <input message="y:getITVByOrgRequest"/>
    <output message="y:getITVByOrgResponse"/>
  </operation>
  <operation name="updateITVRecords">
    <input message="y:updateITVRecordsRequest"/>
    <output message="y:updateITVRecordsResponse"/>
  </operation>
  <operation name="removeITVRecords">
    <input message="y:removeITVRecordsRequest"/>
    <output message="y:removeITVRecordsResponse"/>
  </operation>
</portType>
<binding name="ITVBinding" type="y:ITVPort">
  <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="addITVRecords">
    <soap:operation soapAction="addITVRecords" style="document"/>

```

```

        <input>
          <soap:body parts="request" use="literal"/>
        </input>
      </output>
      <soap:body parts="response" use="literal"/>
    </output>
  </operation>
</operation name="getITVbyId">
  <soap:operation soapAction="getITVbyId" style="document"/>
  <input>
    <soap:body parts="request" use="literal"/>
  </input>
  <output>
    <soap:body parts="response" use="literal"/>
  </output>
</operation>
</operation name="getITVbyNSN">
  <soap:operation soapAction="getITVbyNSN" style="document"/>
  <input>
    <soap:body parts="request" use="literal"/>
  </input>
  <output>
    <soap:body parts="response" use="literal"/>
  </output>
</operation>
</operation name="getITVbyPOD">
  <soap:operation soapAction="getITVbyPOD" style="document"/>
  <input>
    <soap:body parts="request" use="literal"/>
  </input>
  <output>
    <soap:body parts="response" use="literal"/>
  </output>
</operation>
</operation name="getITVbyPOE">
  <soap:operation soapAction="getITVbyPOE" style="document"/>
  <input>
    <soap:body parts="request" use="literal"/>
  </input>
  <output>
    <soap:body parts="response" use="literal"/>
  </output>
</operation>
</operation name="getITVbyNomancature">
  <soap:operation soapAction="getITVbyNomancature" style="document"/>
  <input>
    <soap:body parts="request" use="literal"/>
  </input>
  <output>
    <soap:body parts="response" use="literal"/>
  </output>
</operation>
</operation name="getITVbyOrigin">
  <soap:operation soapAction="getITVbyOrigin" style="document"/>
  <input>
    <soap:body parts="request" use="literal"/>
  </input>
  <output>
    <soap:body parts="response" use="literal"/>
  </output>
</operation>
</operation name="getITVbyDestination">
  <soap:operation soapAction="getITVbyDestination" style="document"/>
  <input>
    <soap:body parts="request" use="literal"/>
  </input>
  <output>
    <soap:body parts="response" use="literal"/>
  </output>
</operation>
</operation name="getITVbySupplyClass">
  <soap:operation soapAction="getITVbySupplyClass" style="document"/>
  <input>
    <soap:body parts="request" use="literal"/>
  </input>
  <output>
    <soap:body parts="response" use="literal"/>
  </output>

```

```

        </output>
    </operation>
    <operation name="getITVbyPerson">
        <soap:operation soapAction="getITVbyPerson" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
    <operation name="getITVbyOrg">
        <soap:operation soapAction="getITVbyOrg" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
    <operation name="updateITVRecords">
        <soap:operation soapAction="updateITVRecords" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
    <operation name="removeITVRecords">
        <soap:operation soapAction="removeITVRecords" style="document"/>
        <input>
            <soap:body parts="request" use="literal"/>
        </input>
        <output>
            <soap:body parts="response" use="literal"/>
        </output>
    </operation>
</binding>
<service name="ITV">
    <port name="ITV" binding="y:ITVBinding">
        <soap:address location="http://sampleLocation.com"/>
    </port>
</service>
</definitions>

```

16 – ITVServiceSOAP

```

<xs:schema targetNamespace="http://coalition/xml/schema"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:cool="http://coalition/xml/schema">
  <xs:include schemaLocation="./common/BaseTypes.xsd"/>
  <xs:element name="booleanResponse" type="xs:boolean">
    <xs:annotation>
      <xs:documentation>A boolean response.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="longResponse" type="xs:long"/>
  <xs:element name="addITVRecordsRequest" type="cool:CargoITVListT"/>
  <xs:element name="getITVByIdRequest" type="cool:IdT"/>
  <xs:element name="getITVByNSNRequest" type="xs:string"/>
  <xs:element name="getITVByPODRequest" type="xs:string"/>
  <xs:element name="getITVByPOERequest" type="xs:string"/>
  <xs:element name="getITVByNomancLatureRequest" type="xs:string"/>
  <xs:element name="getITVByOriginRequest" type="cool:LocationT"/>
  <xs:element name="getITVByDestinationRequest" type="cool:LocationT"/>
  <xs:element name="getITVBySupplyClassRequest" type="xs:string"/>
  <xs:element name="getITVByPersonRequest" type="cool:PersonT"/>
  <xs:element name="getITVByOrgRequest" type="cool:IdT"/>
  <xs:element name="updateITVRecordsRequest" type="cool:CargoITVListT"/>
  <xs:element name="removeITVRecordsRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="cargoId" type="cool:IdT" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="CargoITVList" type="cool:CargoITVListT"/>
  <xs:complexType name="CargoITVListT">
    <xs:sequence>
      <xs:element name="CargoITV" type="cool:CargoITVT" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

17 - CoalitionID

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:y="http://ws.actd.ctl"
  xmlns:cool="http://coalition/xml/schema" targetNamespace="http://ws.actd.ctl">
  <types>
    <xs:schema targetNamespace="http://coalition/xml/schema">
      <xs:import namespace="http://coalition/xml/schema"
        schemaLocation="./coalitionSchema/common/BaseTypes.xsd"/>
    </xs:schema>
  </types>
  <message name="getListRequest">
    <part name="idType" type="xs:string"/>
  </message>
  <message name="getListResponse">
    <part name="startOfIdList" type="cool:IdT"/>
  </message>
  <message name="lookupIdRequest">
    <part name="coalitionId" type="cool:IdT"/>
  </message>
  <message name="lookupIdResponse">
    <part name="idType" type="xs:string"/>
  </message>
  <portType name="CoalitionIDPort">
    <operation name="getList">
      <input message="y:getListRequest"/>
      <output message="y:getListResponse"/>
    </operation>
    <operation name="lookupId">
      <input message="y:lookupIdRequest"/>
      <output message="y:lookupIdResponse"/>
    </operation>
  </portType>
  <binding name="CoalitionIDBinding" type="y:CoalitionIDPort">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="getList">
      <soap:operation soapAction="getList" style="rpc"/>
      <input>
        <soap:body parts="idType" use="literal" namespace="http://ws.actd.ctl"/>
      </input>
      <output>
        <soap:body parts="startOfIdList" use="literal"
          namespace="http://ws.actd.ctl"/>
      </output>
    </operation>
    <operation name="lookupId">
      <soap:operation soapAction="lookupId" style="rpc"/>
      <input>
        <soap:body parts="coalitionId" use="literal"
          namespace="http://ws.actd.ctl"/>
      </input>
      <output>
        <soap:body parts="idType" use="literal" namespace="http://ws.actd.ctl"/>
      </output>
    </operation>
  </binding>
  <service name="CoalitionID">
    <port name="CoalitionID" binding="y:CoalitionIDBinding">
      <soap:address location="http://sampleLocation.com"/>
    </port>
  </service>
</definitions>

```

18 - Language

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:y="http://ws.actd.ctl"
  targetNamespace="http://ws.actd.ctl">
  <types>
    <xs:schema targetNamespace="http://ws.actd.ctl">
      <xs:element name="convertDocumentRequest">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="document" type="xs:anyType"/>
            <xs:element name="sourceLang" type="xs:string"/>
            <xs:element name="targetLang" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="convertDocumentResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="document" type="xs:anyType"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </types>
  <message name="convertDocumentRequest">
    <part name="request" element="y:convertDocumentRequest"/>
  </message>
  <message name="convertDocumentResponse">
    <part name="response" element="y:convertDocumentResponse"/>
  </message>
  <portType name="LanguagePort">
    <operation name="convertDocument">
      <input message="y:convertDocumentRequest"/>
      <output message="y:convertDocumentResponse"/>
    </operation>
  </portType>
  <binding name="LanguageBinding" type="y:LanguagePort">
    <soap:binding style="document"
      transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="convertDocument">
      <soap:operation soapAction="convertDocument" style="document"/>
      <input>
        <soap:body parts="request" use="literal"/>
      </input>
      <output>
        <soap:body parts="response" use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="Language">
    <port name="Language" binding="y:LanguageBinding">
      <soap:address location="http://sampleLocation.com"/>
    </port>
  </service>
</definitions>

```

19 - Unit

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:y="http://ws.actd.ctl"
  targetNamespace="http://ws.actd.ctl">
  <types>
    <xs:schema/>
  </types>
  <message name="convertUnitRequest">
    <part name="value" type="xs:string"/>
    <part name="sourceUnits" type="xs:string"/>
    <part name="targetUnits" type="xs:string"/>
  </message>
  <message name="convertUnitResponse">
    <part name="newValue" type="xs:string"/>
  </message>
  <portType name="UnitPort">
    <operation name="convertUnit">
      <input message="y:convertUnitRequest"/>
      <output message="y:convertUnitResponse"/>
    </operation>
  </portType>
  <binding name="UnitBinding" type="y:UnitPort">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="convertUnit">
      <soap:operation soapAction="convertUnit" style="rpc"/>
      <input>
        <soap:body parts="value sourceUnits targetUnits" use="literal"
namespace="http://ws.actd.ctl"/>
      </input>
      <output>
        <soap:body parts="newValue" use="literal" namespace="http://ws.actd.ctl"/>
      </output>
    </operation>
  </binding>
  <service name="Unit">
    <port name="Unit" binding="y:UnitBinding">
      <soap:address location="hyyp://sampleLocation.com"/>
    </port>
  </service>
</definitions>

```

20 – Location

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:y="http://ws.actd.ctl"
  xmlns:cool="http://coalition/xml/schema" targetNamespace="http://ws.actd.ctl">
  <types>
    <xs:schema targetNamespace="http://coalition/xml/schema">
      <xs:import namespace="http://coalition/xml/schema"
        schemaLocation="../coalitionSchema/LocationServiceSOAP.xsd"/>
    </xs:schema>
  </types>
  <message name="lookupUNLOCRequest">
    <part name="request" element="cool:lookupUNLOCRequest"/>
  </message>
  <message name="lookupUNLOCResponse">
    <part name="response" element="cool:lookupUNLOCResponse"/>
  </message>
  <message name="lookupICA0Request">
    <part name="request" element="cool:lookupICA0Request"/>
  </message>
  <message name="lookupICA0Response">
    <part name="response" element="cool:lookupICA0Response"/>
  </message>
  <message name="lookupPortCodeRequest">
    <part name="request" element="cool:lookupPortCodeRequest"/>
  </message>
  <message name="lookupPortCodeResponse">
    <part name="response" element="cool:lookupPortCodeResponse"/>
  </message>
  <message name="convertToUTMRequest">
    <part name="request" element="cool:convertToUTMRequest"/>
  </message>
  <message name="convertToUTMResponse">
    <part name="response" element="cool:convertToUTMResponse"/>
  </message>
  <message name="convertToLatLonRequest">
    <part name="request" element="cool:convertToLatLonRequest"/>
  </message>
  <message name="convertToLatLonResponse">
    <part name="response" element="cool:convertToLatLonResponse"/>
  </message>
  <message name="getLocationInfoRequest">
    <part name="request" element="cool:getLocationInfoRequest"/>
  </message>
  <message name="getLocationInfoResponse">
    <part name="response" element="cool:getLocationInfoResponse"/>
  </message>
  <portType name="LocationPort">
    <operation name="lookupUNLOC">
      <input message="y:lookupUNLOCRequest"/>
      <output message="y:lookupUNLOCResponse"/>
    </operation>
    <operation name="lookupICA0">
      <input message="y:lookupICA0Request"/>
      <output message="y:lookupICA0Response"/>
    </operation>
    <operation name="lookupPortCode">
      <input message="y:lookupPortCodeRequest"/>
      <output message="y:lookupPortCodeResponse"/>
    </operation>
    <operation name="convertToUTM">
      <input message="y:convertToUTMRequest"/>
      <output message="y:convertToUTMResponse"/>
    </operation>
    <operation name="convertToLatLon">
      <input message="y:convertToLatLonRequest"/>
      <output message="y:convertToLatLonResponse"/>
    </operation>
    <operation name="getLocationInfo">
      <input message="y:getLocationInfoRequest"/>
      <output message="y:getLocationInfoResponse"/>
    </operation>
  </portType>

```

```

</portType>
<binding name="LocationBinding" type="y:LocationPort">
  <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="lookupUNLOC">
    <soap:operation soapAction="lookupUNLOC" style="document"/>
    <input>
      <soap:body parts="request" use="literal"/>
    </input>
    <output>
      <soap:body parts="response" use="literal"/>
    </output>
  </operation>
  <operation name="lookupICA0">
    <soap:operation soapAction="lookupICA0" style="document"/>
    <input>
      <soap:body parts="request" use="literal"/>
    </input>
    <output>
      <soap:body parts="response" use="literal"/>
    </output>
  </operation>
  <operation name="lookupPortCode">
    <soap:operation soapAction="lookupPortCode" style="document"/>
    <input>
      <soap:body parts="request" use="literal"/>
    </input>
    <output>
      <soap:body parts="response" use="literal"/>
    </output>
  </operation>
  <operation name="convertToUTM">
    <soap:operation soapAction="convertToUTM" style="document"/>
    <input>
      <soap:body parts="request" use="literal"/>
    </input>
    <output>
      <soap:body parts="response" use="literal"/>
    </output>
  </operation>
  <operation name="convertToLatLon">
    <soap:operation soapAction="convertToLatLon" style="document"/>
    <input>
      <soap:body parts="request" use="literal"/>
    </input>
    <output>
      <soap:body parts="response" use="literal"/>
    </output>
  </operation>
  <operation name="getLocationInfo">
    <soap:operation soapAction="getLocationInfo" style="document"/>
    <input>
      <soap:body parts="request" use="literal"/>
    </input>
    <output>
      <soap:body parts="response" use="literal"/>
    </output>
  </operation>
</binding>
<service name="Location">
  <port name="Location" binding="y:LocationBinding">
    <soap:address location="http://sampleLocation.com"/>
  </port>
</service>
</definitions>

```

21 – LocationServiceSOAP

```

<xs:schema targetNamespace="http://coalition/xml/schema"
xmlns:cool="http://coalition/xml/schema" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation="./common/BaseTypes.xsd"/>
  <xs:element name="lookupUNLOCRequest" type="cool:LocationT"/>
  <xs:element name="lookupUNLOCResponse" type="xs:string"/>
  <xs:element name="lookupICA0Request" type="cool:LocationT"/>
  <xs:element name="lookupICA0Response" type="xs:string"/>
  <xs:element name="lookupPortCodeRequest" type="cool:LocationT"/>
  <xs:element name="lookupPortCodeResponse" type="xs:string"/>
  <xs:element name="convertToUTMRequest" type="cool:LocationT"/>
  <xs:element name="convertToUTMResponse" type="xs:string"/>
  <xs:element name="convertToLatLonRequest" type="cool:LocationT"/>
  <xs:element name="convertToLatLonResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="lat" type="xs:string"/>
        <xs:element name="lon" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="getLocationInfoRequest" type="cool:IdT"/>
  <xs:element name="getLocationInfoResponse" type="cool:LocationT"/>
</xs:schema>

```


Appendix B: Coalition XML Schema Listing

Below is a complete listing of the coalition XML schema used by the WSDL files and in creating XML documents to be sent to and from all CTL ACTD web services. Electronic copies of all CTL XML Schemas are contained on the CD-ROM attached to this report. If no CD-ROM is present please contact Egon Kuster from Command and Control Division in the Defence Science and Technology Organisation.

```
<xs:schema targetNamespace="http://coalition/xml/schema"
xmlns="http://coalition/xml/schema" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:cool="http://coalition/xml/schema" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="3.2">
  <xs:simpleType name="IdT">
    <xs:annotation>
      <xs:documentation>Defines a generic coalition Id</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:maxLength value="60"/>
      <xs:whiteSpace value="preserve"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="DescriptionT">
    <xs:annotation>
      <xs:documentation>Defines a generic description</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:maxLength value="4000"/>
      <xs:whiteSpace value="preserve"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="GenericParameter50T">
    <xs:annotation>
      <xs:documentation>Defines a generic parameter that is no more than 50
characters</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:maxLength value="50"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="VersionT">
    <xs:annotation>
      <xs:documentation>Defines a generic version type.</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:maxLength value="10"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="ParameterNameT">
    <xs:annotation>
      <xs:documentation>Defines a generic parameter name</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:maxLength value="250"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="BaseNameT">
    <xs:annotation>
      <xs:documentation>Defines a generic name</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:maxLength value="100"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="CarrierSourceT">
    <xs:annotation>
      <xs:documentation>Defines a carrier source. Can be commercial, government,
military or other.</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="COMMERCIAL"/>
      <xs:enumeration value="MILITARY"/>
      <xs:enumeration value="GOVERNMENT"/>
      <xs:enumeration value="OTHER"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

```

    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="CodeT">
    <xs:annotation>
      <xs:documentation>CODETYPE specifies the particular encoding
system.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="Parameter" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>PARAMNAME is the name of the encoding system
parameter; The parameter is actual code value of the parameter
name.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="cool:ParameterNameT">
              <xs:attribute name="parmName" use="required">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:maxLength value="100"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:attribute>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="codeType" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="100"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
  <xs:complexType name="CapabilityT">
    <xs:annotation>
      <xs:documentation>Describes a Capability</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
      <xs:extension base="DescriptionT">
        <xs:attribute name="capabilityId" type="IdT" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="NameT">
    <xs:annotation>
      <xs:documentation>Describes a generic name. ABBR is the abbreviated form of
the name</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
      <xs:extension base="cool:BaseNameT">
        <xs:attribute name="abbr" type="cool:GenericParameter50T" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="PersonNmT">
    <xs:annotation>
      <xs:documentation>Describes a person in the CIE</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="Title" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Dr, Ms, etc.</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:minLength value="0"/>
            <xs:maxLength value="10"/>
            <xs:whiteSpace value="collapse"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Rank" minOccurs="0">

```

```

    <xs:annotation>
      <xs:documentation>National rank information</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:whiteSpace value="preserve"/>
        <xs:maxLength value="100"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="FirstNm">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="50"/>
        <xs:whiteSpace value="preserve"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="MiddleNm" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:minLength value="0"/>
        <xs:maxLength value="20"/>
        <xs:whiteSpace value="preserve"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="LastNm">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="50"/>
        <xs:whiteSpace value="preserve"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="OtherNm" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:minLength value="0"/>
        <xs:maxLength value="50"/>
        <xs:whiteSpace value="preserve"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="Gender" minOccurs="0">
    <xs:annotation>
      <xs:documentation>M or F</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="F"/>
        <xs:enumeration value="M"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="PersonT">
  <xs:annotation>
    <xs:documentation>ID uniquely identifies a person in the coalition domain. It describes a person in the coalition</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="cool:PersonNmT">
      <xs:sequence>
        <xs:element name="Nationality" type="xs:string" minOccurs="0">
          <xs:annotation>
            <xs:documentation>The country of citizenship</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="NationalId" type="xs:string" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Serial Number, Employee Id, Military Id,
etc.</xs:documentation>
          </xs:annotation>

```

```

        </xs:element>
        <xs:element name="OrgId" type="xs:string">
          <xs:annotation>
            <xs:documentation>Although a person who being transported may
not have a country, they are always associated with some
organization.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="BloodType" minOccurs="0">
          <xs:complexType>
            <xs:attribute name="group" use="required">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="A"/>
                  <xs:enumeration value="B"/>
                  <xs:enumeration value="O"/>
                  <xs:enumeration value="AB"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="rhStatus" use="required">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="RHPLUS"/>
                  <xs:enumeration value="RHMINUS"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </xs:complexType>
        </xs:element>
        <xs:element name="Religion" minOccurs="0">
          <xs:annotation>
            <xs:documentation>This person's religious
affiliation.</xs:documentation>
          </xs:annotation>
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:whiteSpace value="preserve"/>
              <xs:maxLength value="100"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="Occupation" minOccurs="0">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="1000"/>
              <xs:whiteSpace value="preserve"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="Language" minOccurs="0" maxOccurs="unbounded">
          <xs:annotation>
            <xs:documentation>LANGCODE is the ISO-639-2 Language
Identification. MOSTFLUENT attribute =1 is a person's primary language; most fluent =
2 secondary and so forth.</xs:documentation>
          </xs:annotation>
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="cool:LanguageT">
                <xs:attribute name="mostFluent" use="required">
                  <xs:simpleType>
                    <xs:restriction base="xs:integer">
                      <xs:minInclusive value="1"/>
                      <xs:whiteSpace value="collapse"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:attribute>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
        <xs:element name="PointOfContact" type="cool:PointOfContactT"
minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="personId" type="IdT" use="required"/>
    </xs:extension>

```

```

    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="PersonTransT">
    <xs:annotation>
      <xs:documentation>PERSONID uniquely identifies a person in the coalition;
      TRANSITID is a unique tracking number in the coalition domain that can be mapped to on
      or more national domain tracking numbers. </xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="PersonT">
        <xs:sequence>
          <xs:element ref="TransitNum"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="LanguageT">
    <xs:annotation>
      <xs:documentation>ISO 639-2 Language Identification LANGCODE is the 639-2 3-
      character code that corresponds to a language. For example for NEP for
      Nepali.</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
      <xs:extension base="cool:BaseNameT">
        <xs:attribute name="langCode" use="required">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="3"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="OrgStructT">
    <xs:annotation>
      <xs:documentation>Defines a generic mechanism to describe an organization
      structure. Allows different hierarchies to be defined for the same
      data</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="OrgStructure" type="OrgStructT" minOccurs="0"
      maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="orgId" type="IdT" use="required"/>
  </xs:complexType>
  <xs:complexType name="HierarchyT">
    <xs:sequence>
      <xs:element name="OrganizationStructure" type="cool:OrgStructT"/>
      <xs:element name="Desc" type="cool:DescriptionT" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Description of the organization
          structure</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="name" type="BaseNameT" use="required"/>
  </xs:complexType>
  <xs:complexType name="OrgInfoT">
    <xs:annotation>
      <xs:documentation>An organization in the CIE</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="Name" type="cool:NameT"/>
      <xs:element name="NationalId" minOccurs="0">
        <xs:annotation>
          <xs:documentation>National domain identification
          code</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="idType" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="100"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>

```

```

        </xs:simpleType>
      </xs:attribute>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="Echelon" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Battalion, Brigade, etc.</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="BRIGADE"/>
      <xs:enumeration value="BATTALION"/>
      <xs:enumeration value="BRANCH"/>
      <xs:enumeration value="BATTERY"/>
      <xs:enumeration value="COMMAND"/>
      <xs:enumeration value="COMPANY"/>
      <xs:enumeration value="CORPS"/>
      <xs:enumeration value="CREW"/>
      <xs:enumeration value="DIVISION"/>
      <xs:enumeration value="DETACHMENT"/>
      <xs:enumeration value="DETAIL"/>
      <xs:enumeration value="ELEMENT"/>
      <xs:enumeration value="FORCE"/>
      <xs:enumeration value="FLIGHT"/>
      <xs:enumeration value="GROUP"/>
      <xs:enumeration value="HEADQUARTERS (HQ), HQ COMPANY AND BAND"/>
      <xs:enumeration value="HQ AND HQ BATTERY"/>
      <xs:enumeration value="HQ AND HQ COMPANY"/>
      <xs:enumeration value="HQ AND HQ DETACHMENT"/>
      <xs:enumeration value="HQ, HQ AND COMPANY, AND SERVICE COMPANY"/>
      <xs:enumeration value="HQ AND HQ TROOP"/>
      <xs:enumeration value="HOME"/>
      <xs:enumeration value="HQ AND MAINTENANCE COMPANY"/>
      <xs:enumeration value="HEADQUARTERS"/>
      <xs:enumeration value="HEADQUARTERS COMPANY"/>
      <xs:enumeration value="HEADQUARTERS DETACHMENT"/>
      <xs:enumeration value="HEADQUARTERS AND SERVICE COMPANY"/>
      <xs:enumeration value="HQ, HQ AND SERVICE BATTERY"/>
      <xs:enumeration value="HQ, HQ AND SUPPORT COMPANY"/>
      <xs:enumeration value="PLATOON"/>
      <xs:enumeration value="PARTY"/>
      <xs:enumeration value="REGIMENT"/>
      <xs:enumeration value="SERVICE COMPANY"/>
      <xs:enumeration value="SECTION"/>
      <xs:enumeration value="SQUADRON"/>
      <xs:enumeration value="SQUAD"/>
      <xs:enumeration value="STAFF"/>
      <xs:enumeration value="SPECIAL TROOPS"/>
      <xs:enumeration value="TASK ELEMENT"/>
      <xs:enumeration value="TASK FORCE"/>
      <xs:enumeration value="TEAM"/>
      <xs:enumeration value="TROOP"/>
      <xs:enumeration value="UNIT"/>
      <xs:enumeration value="WING"/>
      <xs:enumeration value="FLEET"/>
      <xs:enumeration value="FLOTILLA"/>
      <xs:enumeration value="WATCH"/>
      <xs:enumeration value="UNKNOWN"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Version" type="VersionT" minOccurs="0"/>
<xs:element name="FuncDesc" type="cool:DescriptionT" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Functional Description. What the organization
does</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="Mission" type="cool:DescriptionT" minOccurs="0">
  <xs:annotation>
    <xs:documentation>The organization's coalition mission or coalition
assignment</xs:documentation>
  </xs:annotation>
</xs:element>

```

```

<xs:element name="MissionConst" type="cool:DescriptionT" minOccurs="0">
  <xs:annotation>
    <xs:documentation>A textual description of restrictions (mission
constraints) concerning how this organization can be utilized in the coalition
domain</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="Authorization" type="cool:DescriptionT" minOccurs="0">
  <xs:annotation>
    <xs:documentation>The authorization that allows the organization to
operate in or participate with the coalition.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="CountrySt" type="cool:LocationT" minOccurs="0">
  <xs:annotation>
    <xs:documentation>The country/state that this organization is
associated with. If the organization is sovereign state, this is the name of the
state</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="OrgType">
  <xs:complexType>
    <xs:choice>
      <xs:element name="Military">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ForceClassification">
              <xs:annotation>
                <xs:documentation>This describes the basic type of
force (AIR, LAND, SEA or SPACE).</xs:documentation>
              </xs:annotation>
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="AIR"/>
                  <xs:enumeration value="LAND"/>
                  <xs:enumeration value="SEA"/>
                  <xs:enumeration value="SPACE"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="NationalService" minOccurs="0">
              <xs:annotation>
                <xs:documentation>Nation domain service name for
example Airforce, Marines etc.</xs:documentation>
              </xs:annotation>
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="AIR_FORCE"/>
                  <xs:enumeration value="ARMY"/>
                  <xs:enumeration value="NAVY"/>
                  <xs:enumeration value="MARINE"/>
                  <xs:enumeration value="COAST_GUARD"/>
                  <xs:enumeration value="JOINT"/>
                  <xs:enumeration value="OTHER"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="MilitaryOrg">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="ADMINISTRATIVE"/>
                  <xs:enumeration value="AIRCRAFT DEVELOPMENT"/>
                  <xs:enumeration value="ARMOR, ANTITANK, TRACKED
VEHICLES"/>
                  <xs:enumeration value="ARTILLERY, AIR DEFENSE
MISSILES"/>
                  <xs:enumeration value="AVIATION
SUPPORT/MAINTENANCE"/>
                  <xs:enumeration value="AVN FLT UNITS, MISSION
AIRCRAFT"/>
                  <xs:enumeration value="BRIDGE LAYING"/>
                  <xs:enumeration value="CHEMICAL, ORDNANCE"/>
                  <xs:enumeration value="COMBAT SERVICE SUPPORT"/>
                  <xs:enumeration value="COMMAND HEADQUARTERS,
DEFENSE AGENCIES"/>
                  <xs:enumeration value="COMMUNICATIONS,
ELECTRONICS, SIGNAL"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:choice>
  </xs:complexType>
</xs:element>

```

```

ENTITIES"/>
    <xs:enumeration value="CONSTRUCTION"/>
    <xs:enumeration value="DEFENSE/CIVIL GOVT

SPECIFIED"/>
    <xs:enumeration value="ELECTRONIC WARFARE"/>
    <xs:enumeration value="ELECTRONICS"/>
    <xs:enumeration value="ENGINEER, ARMoured"/>
    <xs:enumeration value="ENGINEER, NOT OTHERWISE

DISPOSAL"/>
    <xs:enumeration value="ENGINEER, TOPO SERVICE"/>
    <xs:enumeration value="EXPLOSIVE ORDNANCE

AFFAIRS"/>
    <xs:enumeration value="FLT AUXIL ADMIN"/>
    <xs:enumeration value="HEADQUARTERS SUPPORT"/>
    <xs:enumeration value="HUMANITARIAN"/>
    <xs:enumeration value="INFANTRY"/>
    <xs:enumeration value="INTELLIGENCE"/>
    <xs:enumeration value="LABOUR RESOURCES"/>
    <xs:enumeration value="MAINTENANCE"/>
    <xs:enumeration value="MEDICAL"/>
    <xs:enumeration value="MEDICAL ARMoured"/>
    <xs:enumeration value="MEDICAL, DENTAL"/>
    <xs:enumeration value="METEOROLOGICAL"/>
    <xs:enumeration value="MILITARY CIVILIAN

SERV SPT, UNIT HQ"/>
    <xs:enumeration value="MILITARY POLICE"/>
    <xs:enumeration value="MINEFIELD"/>
    <xs:enumeration value="MISC, CMBT, CMBT SPT, CMBT

HOLDING UNIT"/>
    <xs:enumeration value="MULTIROLE"/>
    <xs:enumeration value="NAVAL SUPPORT ELEMENT"/>
    <xs:enumeration value="NOT KNOWN"/>
    <xs:enumeration value="ORDNANCE"/>
    <xs:enumeration value="OTHER"/>
    <xs:enumeration value="PAY/FINANCE"/>
    <xs:enumeration value="PERSONNEL SERVICES"/>
    <xs:enumeration value="PIPELINE"/>
    <xs:enumeration value="POSTAL AND COURIER"/>
    <xs:enumeration value="PSYCHOLOGICAL"/>
    <xs:enumeration value="QUARTERMASTER/LOGISTICS"/>
    <xs:enumeration value="RANGING, ELECTRONIC"/>
    <xs:enumeration value="REINFORCEMENT/REPLACEMENT

TEST AND EVALUATION"/>
    <xs:enumeration value="RESEARCH, DEVELOPMENT,

RESCUE"/>
    <xs:enumeration value="ROAD/RAILWAY"/>
    <xs:enumeration value="SHIPPING"/>
    <xs:enumeration value="SIGNALS"/>
    <xs:enumeration value="SOUND RANGING"/>
    <xs:enumeration value="SUPPLY"/>
    <xs:enumeration value="SURVEY"/>
    <xs:enumeration value="TACTICAL CONTROL, WEATHER

NAVY MOBILE UNITS, AVN"/>
    <xs:enumeration value="TASK ORGANIZATION"/>
    <xs:enumeration value="TRAINING"/>
    <xs:enumeration value="TRANSPORTATION"/>
    <xs:enumeration value="TRANSPORTATION MEDICAL"/>
    <xs:enumeration value="TRANSPORTATION MOVEMENT"/>
    <xs:enumeration value="UNCONVENTIONAL WARFARE,

TRAINING"/>
    <xs:enumeration value="VETERINARY"/>
    <xs:enumeration value="WARSHIPS, CRAFT ADMIN, AVN

    </xs:restriction>
    </xs:simpleType>
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="NonMilitary">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="NonMilitaryOrg">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="HUMANITARIAN"/>
            <xs:enumeration value="TRANSPORTATION"/>

```

```

        <xs:enumeration value="OTHER"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
<xs:element name="Classification">
  <xs:annotation>
    <xs:documentation>Organizations are classified at a high-level as one
of the following types: Coalition, Coalition Partner or Non-
coalition</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="COALITION"/>
      <xs:enumeration value="COALITION_PARTNER"/>
      <xs:enumeration value="NON_COALITION"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Capability" type="cool:CapabilityT" minOccurs="0"
maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>A list of organization
capabilities</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="Person" type="cool:PersonT" minOccurs="0"
maxOccurs="unbounded"/>
<xs:element name="NumPeople" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Language" type="cool:LanguageT" minOccurs="0">
  <xs:annotation>
    <xs:documentation>If the organization is a coalition partner, this is
the primary language spoken. If this is a coalition, this is the agreed official
language used to communicate in the coalition. </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="DefMeasSys" minOccurs="0">
  <xs:annotation>
    <xs:documentation>The default measurement system associated with this
organization</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="ENGLISH"/>
      <xs:enumeration value="METRIC"/>
      <xs:enumeration value="OTHER"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
</xs:sequence>
<xs:attribute name="orgId" type="IdT" use="required"/>
</xs:complexType>
<xs:complexType name="OrgEquipT">
  <xs:annotation>
    <xs:documentation>An Organization and its equipment</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="cool:OrgInfoT">
      <xs:sequence minOccurs="0">
        <xs:element name="Equipment" type="cool:BaseEquipT" minOccurs="0"
maxOccurs="unbounded">
          <xs:annotation>
            <xs:documentation>EQUIPMENTID uniquely identifies a piece of
equipment in the coalition domain. </xs:documentation>
          </xs:annotation>
        </xs:element>

```

```

        <xs:element name="Supply" type="BaseSupplyItemT" minOccurs="0"
maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>SUPPLYID uniquely identifies a supply item in
the coalition domain. </xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Container" type="EmptyContainerT" minOccurs="0"
maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>Describes a container (or a pallet).
CONTAINERID uniquely identifies a container using any international standard such as
ISO6346, ANSI/UCC6 (SSCC), etc. CONTAINERIDTYPE defines the type of container
Id.</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="OrgEquipTransT">
    <xs:annotation>
        <xs:documentation>An Organization, equipment and material that is being
transported.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="TransitNum" type="TransitIdT" maxOccurs="unbounded"/>
        <xs:element name="OrgEquip" type="OrgEquipT" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="LocationT">
    <xs:annotation>
        <xs:documentation>ID uniquely identifies a location in the coalition domain.
It describes an approved location code</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="Name" type="cool:NameT"/>
        <xs:element name="Code" maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>CODETYPE could be LATLON, ICAO, IATA, ISO 3166,
UNLOCODE, GEOLOC etc. ; </xs:documentation>
            </xs:annotation>
            <xs:complexType>
                <xs:complexContent>
                    <xs:extension base="CodeT">
                        <xs:sequence minOccurs="0">
                            <xs:element name="Desc" type="cool:DescriptionT"
minOccurs="0"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
        </xs:element>
        <xs:element name="Address" type="ConsignorT" minOccurs="0">
            <xs:annotation>
                <xs:documentation>addId is a uniquely identifies a location address.
This usually an address id for a DODAAC.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Desc" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Textual description of the
location</xs:documentation>
            </xs:annotation>
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="4000"/>
                    <xs:whiteSpace value="preserve"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="TimeZone" minOccurs="0">
            <xs:annotation>
                <xs:documentation>This is an offset relative to GMT time. Each time
zone is either a single letter (integral offset) or a letter and a non-letter (i.e.
non-integral offset 3:30 hours, 11:30 hours, etc.)</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:complexType>

```

```

</xs:annotation>
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:enumeration value="0"/>
    <xs:enumeration value="+1"/>
    <xs:enumeration value="+2"/>
    <xs:enumeration value="+3"/>
    <xs:enumeration value="+3:30"/>
    <xs:enumeration value="+4"/>
    <xs:enumeration value="+4:30"/>
    <xs:enumeration value="+5"/>
    <xs:enumeration value="+5:30"/>
    <xs:enumeration value="+6"/>
    <xs:enumeration value="+6:30"/>
    <xs:enumeration value="+7"/>
    <xs:enumeration value="+8"/>
    <xs:enumeration value="+9"/>
    <xs:enumeration value="+9:30"/>
    <xs:enumeration value="+10"/>
    <xs:enumeration value="+10:30"/>
    <xs:enumeration value="+11"/>
    <xs:enumeration value="+11:30"/>
    <xs:enumeration value="+12"/>
    <xs:enumeration value="+13"/>
    <xs:enumeration value="+14"/>
    <xs:enumeration value="-1"/>
    <xs:enumeration value="-2"/>
    <xs:enumeration value="-3"/>
    <xs:enumeration value="-3:30"/>
    <xs:enumeration value="-4"/>
    <xs:enumeration value="-5"/>
    <xs:enumeration value="-6"/>
    <xs:enumeration value="-7"/>
    <xs:enumeration value="-8"/>
    <xs:enumeration value="-8:30"/>
    <xs:enumeration value="-9"/>
    <xs:enumeration value="-9:30"/>
    <xs:enumeration value="-10"/>
    <xs:enumeration value="-11"/>
    <xs:enumeration value="-12"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>
</xs:sequence>
<xs:attribute name="locationId" type="IdT" use="required"/>
</xs:complexType>
<xs:complexType name="BaseEquipT">
  <xs:annotation>
    <xs:documentation>Describes a generic type of equipment. EQUIPID uniquely
identifies a piece of equipment in the coalition domain (i.e. a Nato Stock Number).
</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Name" type="cool:NameT">
      <xs:annotation>
        <xs:documentation>Equipment name</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Code">
      <xs:annotation>
        <xs:documentation>National Domain Id and Id type</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="BaseNameT">
            <xs:attribute name="codeType" type="BaseNameT" use="required"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="Desc" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Textual description of the
equipment</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:string">

```

```

        <xs:maxLength value="4000"/>
        <xs:whiteSpace value="preserve"/>
    </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="Length" type="cool:LengthT"/>
<xs:element name="Height" type="cool:LengthT"/>
<xs:element name="Width" type="cool:LengthT"/>
<xs:element name="Weight" type="cool:WeightT"/>
<xs:element name="Volume" type="cool:VolumeT" minOccurs="0"/>
<xs:element ref="cool:CargoCode" minOccurs="0">
    <xs:annotation>
        <xs:documentation>A coalition domain code that defines categories of
equipment. Based on US domain cargo codes.</xs:documentation>
        <xs:documentation>A coalition domain code that defines categories of
equipment. Based on US domain cargo codes.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="HazCargo" minOccurs="0">
    <xs:annotation>
        <xs:documentation>Hazardous Cargo code.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="codeType" type="xs:string" use="required"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
<xs:element name="NetExplosiveQuantity" type="xs:string" minOccurs="0"/>
</xs:sequence>
<xs:attribute name="equipId" type="IdT" use="required">
    <!-- <xs:annotation>
        <xs:documentation source="Owning organization"/>
    </xs:annotation> -->
</xs:attribute>
</xs:complexType>
<xs:complexType name="EquipShipT">
    <xs:annotation>
        <xs:documentation>EQUIPID uniquely identifies a piece of equipment in the
coalition domain. </xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="cool:BaseEquipT">
            <xs:sequence minOccurs="0">
                <xs:element name="OwningOrgId" type="xs:string"/>
                <xs:element name="Quantity">
                    <xs:simpleType>
                        <xs:restriction base="xs:integer">
                            <xs:minInclusive value="1"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="SpecialHandling" minOccurs="0">
                    <xs:annotation>
                        <xs:documentation>Special handling
instructions</xs:documentation>
                    </xs:annotation>
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:minLength value="0"/>
                            <xs:maxLength value="100"/>
                            <xs:whiteSpace value="preserve"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="Configuration" type="DescriptionT" minOccurs="0">
                    <xs:annotation>
                        <xs:documentation>Shipping configuration of the
equipment</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>

```

```

</xs:complexType>
<xs:complexType name="BaseContainerT">
  <xs:annotation>
    <xs:documentation>Describes a generic container (or a pallet) not associated
with an organization. CONTAINERID uniquely identifies a container using any
international standard such as ISO6346, ANSI/UCC6 (SSCC), etc. CONTAINERIDTYPE defines
the type of container Id.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Length" type="cool:LengthT"/>
    <xs:element name="Height" type="cool:LengthT"/>
    <xs:element name="Width" type="cool:LengthT"/>
    <xs:element name="Weight" type="cool:WeightT"/>
    <xs:element name="HazCargo" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Hazardous Cargo code.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="codeType" use="required">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:maxLength value="50"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="NetExplosiveQuantity" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="100"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Quantity">
      <xs:annotation>
        <xs:documentation>This is the number of items in the container. For
example the number of sub-containers, or pieces of equipment that are not in a sub-
container. A value of zero indicates an empty container.</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:integer">
          <xs:minInclusive value="0"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Equipment" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="EquipShipT"/>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="Supply" minOccurs="0" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>This is a list of supplies</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="SupplyItemT"/>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="Container" type="BaseContainerT" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Desc" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The description of this
container</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="4000"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

```

        <xs:whiteSpace value="preserve"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
</xs:sequence>
<xs:attribute name="containerId" type="IdT" use="required"/>
<xs:attribute name="containerIdType" use="required">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="100"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
</xs:complexType>
<xs:complexType name="EquipTransT">
  <xs:annotation>
    <xs:documentation>EQUIPID uniquely identifies a piece of equipment in the
coalition domain; TRANSITID is a unique tracking number in the coalition domain that
can be mapped to on or more national domain tracking numbers. </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="EquipShipT">
      <xs:sequence>
        <xs:element name="Consignor" type="ConsignorT" minOccurs="0"/>
        <xs:element name="Consignee" type="ConsignorT" minOccurs="0"/>
        <xs:element ref="TransitNum"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="EmptyContainerT">
  <xs:sequence>
    <xs:element name="Length" type="cool:LengthT"/>
    <xs:element name="Height" type="cool:LengthT"/>
    <xs:element name="Width" type="cool:LengthT"/>
    <xs:element name="Weight" type="cool:WeightT"/>
    <xs:element name="Desc" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The description of this
container</xs:documentation>
      </xs:annotation>
    </xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="4000"/>
      <xs:whiteSpace value="preserve"/>
    </xs:restriction>
  </xs:simpleType>
</xs:sequence>
<!-- <xs:complexContent>
<xs:restriction base="BaseContainerT"> -->
<!-- </xs:restriction>
</xs:complexContent> -->
</xs:complexType>
<xs:complexType name="ContainerT">
  <xs:annotation>
    <xs:documentation>Describes a container (or a pallet). CONTAINERID uniquely
identifies a container using any international standard such as ISO6346, ANSI/UCC6
(SSCC), etc. CONTAINERIDTYPE defines the type of container Id. OWNINGORGID the
coalition identifier that defines the owning organization.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="ContainerIdType">
      <xs:annotation>
        <xs:documentation>Defines a type of container or a pallet
id.</xs:documentation>
      </xs:annotation>
    </xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="ISO_6346"/>
      <xs:enumeration value="BIC"/>
      <xs:enumeration value="UNCE_8260"/>
      <xs:enumeration value="OTHER"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

```

```

<xs:element name="Length" type="cool:LengthT"/>
<xs:element name="Height" type="cool:LengthT"/>
<xs:element name="Width" type="cool:LengthT"/>
<xs:element name="Weight" type="cool:WeightT"/>
<xs:element name="HazCargo" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Hazardous Cargo code.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="codeType" use="required">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="50"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="NetExplosiveQuantity" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="100"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Quantity">
  <xs:annotation>
    <xs:documentation>This is the number of items in the container. For
example the number of sub-containers, or pieces of equipment that are not in a sub-
container. A value of zero indicates an empty container.</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="OwningOrg" type="xs:string">
  <xs:annotation>
    <xs:documentation>OrgId of an Organization that owns this
container.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="Equipment" minOccurs="0" maxOccurs="unbounded">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="EquipShipT"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:element name="Supply" minOccurs="0" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>This is a list of supplies</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="SupplyItemT"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:element name="Container" type="ContainerT" minOccurs="0"
maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>Subcontainer of this container.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="Desc" minOccurs="0">
  <xs:annotation>
    <xs:documentation>The description of this
container</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">

```

```

        <xs:maxLength value="4000"/>
        <xs:whiteSpace value="preserve"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
</xs:sequence>
<xs:attribute name="containerId" type="IdT" use="required"/>
</xs:complexType>
<xs:complexType name="ContainerTransT">
  <xs:annotation>
    <xs:documentation>A container or containers that are being
transported</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="ContainerT">
      <xs:sequence>
        <xs:element ref="TransitNum"/>
        <xs:element name="Consignor" type="ConsignorT" minOccurs="0"/>
        <xs:element name="Consignee" type="ConsignorT" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="ContainerTransReqT">
  <xs:sequence>
    <xs:element name="Length" type="cool:LengthT"/>
    <xs:element name="Height" type="cool:LengthT"/>
    <xs:element name="Width" type="cool:LengthT"/>
    <xs:element name="Weight" type="cool:WeightT"/>
    <xs:element name="HazCargo" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Hazardous Cargo code.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="codeType" use="required">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:maxLength value="50"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="NetExplosiveQuantity" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="100"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Quantity">
      <xs:annotation>
        <xs:documentation>This is the number of items in the container. For
example the number of sub-containers, or pieces of equipment that are not in a sub-
container. A value of zero indicates an empty container.</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:integer">
          <xs:minInclusive value="0"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Equipment" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="cool:EquipTransT">
            <xs:attribute name="reqId" type="IdT" use="optional"/>
            <xs:attribute name="reqIdVersion" use="optional">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:maxLength value="5"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>

```

```

        </xs:simpleType>
      </xs:attribute>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
</xs:element>
<xs:element name="Supply" minOccurs="0" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>This is a list of supplies</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="cool:SupplyTransT">
        <xs:attribute name="reqId" type="IdT" use="optional"/>
        <xs:attribute name="reqIdVersion" use="optional">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="5"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:element name="Container" type="cool:ContainerTransReqT" minOccurs="0"
maxOccurs="unbounded"/>
  <xs:element name="Desc" minOccurs="0">
    <xs:annotation>
      <xs:documentation>The description of this
container</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="4000"/>
        <xs:whiteSpace value="preserve"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="Consignor" type="ConsignorT" minOccurs="0">
    <xs:annotation>
      <xs:documentation>Where the items are being sent. ORGID (optional) the
sending organization.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="Consignee" type="ConsignorT" minOccurs="0">
    <xs:annotation>
      <xs:documentation>Where the items are being sent. ORGID (optional) the
receiving organization.</xs:documentation>
    </xs:annotation>
  </xs:element>
</xs:sequence>
<xs:attribute name="containerId" type="IdT" use="required"/>
<xs:attribute name="containerIdType" use="required">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="100"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="transitId" type="IdT" use="required"/>
<xs:attribute name="reqId" type="IdT" use="optional"/>
<xs:attribute name="reqIdVersion" use="optional">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="5"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
</xs:complexType>
<xs:complexType name="CarrierT">
  <xs:annotation>
    <xs:documentation>Describes a carrier. CARRIERID uniquely identifies a
carrier in the coalition domain.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Name" type="cool:NameT"/>

```

```

<xs:element name="CarrierType" minOccurs="0">
  <xs:annotation>
    <xs:documentation>UN/CEFACT Recommendation 19, or CJ130, etc. TRANSMODE
is air, land_rail, land_motor, sea, space or other; OTHERTRANSDISC is a description of
the type of transportation when the mode is other.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="cool:BaseNameT">
        <xs:attribute name="transMode" use="required">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="10"/>
              <xs:enumeration value="AIR"/>
              <xs:enumeration value="SEA"/>
              <xs:enumeration value="LAND_MOTOR"/>
              <xs:enumeration value="LAND_RAIL"/>
              <xs:enumeration value="SPACE"/>
              <xs:enumeration value="OTHER"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="otherTransDesc" use="optional">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="100"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="CarrierTypeNum" type="xs:string" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Carrier Type Number. If Trans_Mode is air, the
Carrier is an airplane, the CarrierType is AircraftType, the CarrierTypeNum is
AircraftTailNum.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="OrgId" type="xs:string">
  <xs:annotation>
    <xs:documentation>Identifies the owner of the
carrier</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="Operator" minOccurs="0">
  <xs:annotation>
    <xs:documentation>The organization that is under contract to operate
this carrier</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="100"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Source" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Military, commercial, other or government.
</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="cool:CarrierSourceT"/>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="SourceId" minOccurs="0" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>The carrier id or ids that are associated with the
source; SOURCEIDTYPE specifies the type of source id. For example if the source is
commercial, the SOURCEID might be 8025472 and the SOURCEIDTYPE might be IMO Number.
</xs:documentation>
  </xs:annotation>
  <xs:complexType>

```

```

        <xs:simpleContent>
          <xs:extension base="IdT">
            <xs:attribute name="sourceIdType" use="required">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:maxLength value="100"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="Capability" type="CapabilityT" minOccurs="0"
maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>A list of capabilities</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Desc" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="4000"/>
          <xs:whiteSpace value="preserve"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="carrierId" type="IdT" use="required"/>
</xs:complexType>
<xs:complexType name="CarrierManRecT">
  <xs:annotation>
    <xs:documentation>carrierManId-uniquely identifies a line in the carrier
manifest; RECTYPE is 'A' for an allocation record, 'M' for a manifest record or 'B'
for both</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element ref="TransitNum" maxOccurs="unbounded"/>
    <xs:element name="Organization" minOccurs="0" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>List of organizations or sub-organizations (possibly
to individual people)</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="OrgEquipT">
            <xs:attribute name="reqId" type="IdT" use="optional"/>
            <xs:attribute name="reqIdVersion" type="xs:string"
use="optional"/>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="Equipment" minOccurs="0" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>This is equipment that is on the carrier that is not
in a container</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="EquipShipT">
            <xs:sequence>
              <xs:element name="Consignor" type="ConsignorT"
minOccurs="0"/>
              <xs:element name="Consignee" type="ConsignorT"
minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="reqId" type="IdT" use="optional"/>
            <xs:attribute name="reqIdVersion" type="xs:string"
use="optional"/>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="Supply" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:complexContent>

```

```

        <xs:extension base="SupplyItemT">
            <xs:sequence>
                <xs:element name="Consignor" type="ConsignorT"
minOccurs="0"/>
                <xs:element name="Consignee" type="ConsignorT"
minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="reqId" type="IdT" use="optional"/>
            <xs:attribute name="reqIdVersion" type="xs:string"
use="optional"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
</xs:element>
<xs:element name="Container" minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
        <xs:documentation>List of containerized equipment, empty container(s)
needing redeployment</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:complexContent>
            <xs:extension base="ContainerT">
                <xs:sequence>
                    <xs:element name="Consignor" type="ConsignorT"
minOccurs="0"/>
                    <xs:element name="Consignee" type="ConsignorT"
minOccurs="0"/>
                </xs:sequence>
                <xs:attribute name="reqId" type="IdT" use="optional"/>
                <xs:attribute name="reqIdVersion" type="xs:string"
use="optional"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
</xs:element>
<xs:element name="Person" minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
        <xs:documentation>A list of personel </xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:complexContent>
            <xs:extension base="PersonT">
                <xs:attribute name="reqId" type="IdT" use="optional"/>
                <xs:attribute name="reqIdVersion" type="xs:string"
use="optional"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
</xs:element>
<xs:element name="MovePlanId" minOccurs="0">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="IdT">
                <xs:attribute name="version" use="required">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:maxLength value="5"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:attribute>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
<xs:element name="Desc" minOccurs="0">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="4000"/>
            <xs:whiteSpace value="preserve"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
</xs:sequence>
<xs:attribute name="carrierManId" type="IdT" use="required"/>
<xs:attribute name="recType" use="required">
    <xs:simpleType>

```

```

        <xs:restriction base="xs:string">
            <xs:enumeration value="A"/>
            <xs:enumeration value="M"/>
            <xs:enumeration value="B"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
</xs:complexType>
<xs:complexType name="CarrierManT">
    <xs:annotation>
        <xs:documentation>CARRIERID uniquely identifies a carrier in the
coalition;MISSIONVOYNUM could be a waybill number, a commercial carrier tracking
number, bill of lading number etc; </xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="ManifestItem" type="CarrierManRecT" maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>carrierManId -uniquely identifies a line in the
carrier manifest; carrierSchedId uniquely identifies the line in the carrier schedule
associated with this item in the manifest; RECTYPE is 'A' for an allocation record,
'M' for a manifest record or 'B' for both</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="carrierId" type="IdT" use="required"/>
    <xs:attribute name="missionVoyNum" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="30"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
<xs:complexType name="CarrierSchedItemT">
    <xs:annotation>
        <xs:documentation>carrierSchedId uniquely identifies a line in the carrier
schedule.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="Departure" type="cool:DateTimeLocT"/>
        <xs:element name="Arrival" type="cool:DateTimeLocT"/>
    </xs:sequence>
    <xs:attribute name="carrierSchedId" type="IdT" use="required"/>
    <xs:attribute name="missionVoyNum" type="xs:string" use="required"/>
</xs:complexType>
<xs:complexType name="CarrierSchedT">
    <xs:annotation>
        <xs:documentation>carrierSchedId uniquely identifies a line in the carrier
schedule; MISSIONVOYAGE could be a waybill number, a commercial carrier tracking
number, bill of lading number etc</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="CarrierSchedItemT">
            <xs:sequence>
                <xs:element name="ManifestItem" type="cool:CarrierManRecT"
minOccurs="0" maxOccurs="unbounded">
                    <xs:annotation>
                        <xs:documentation>carrierManId-uniquely identifies a line in the
carrier manifest; RECTYPE is 'A' for an allocation record, 'M' for a manifest
record or 'B' for both</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="DateTimeT">
    <xs:annotation>
        <xs:documentation>Defines a date time type, that specifies a date/time
relative to the commencement day</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute name="format" use="required">
                <xs:simpleType>
                    <xs:restriction base="xs:string"/>
                </xs:simpleType>
            </xs:attribute>
        </xs:extension>
    </xs:simpleContent>

```

```

        </xs:attribute>
    </xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="DateTimeLocT">
    <xs:annotation>
        <xs:documentation>Defines a scheduled and actual day/time as well as a
location type</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="Scheduled" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Scheduled location, date and
time.</xs:documentation>
            </xs:annotation>
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="Location" type="LocationT" minOccurs="0"/>
                    <xs:element name="DateTime" type="cool:DateTimeT"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element name="Actual" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Actual location date and time.</xs:documentation>
            </xs:annotation>
            <xs:complexType>
                <xs:sequence minOccurs="0">
                    <xs:element name="Location" type="LocationT" minOccurs="0"/>
                    <xs:element name="DateTime" type="cool:DateTimeT"/>
                    <xs:element name="Delay" minOccurs="0">
                        <xs:annotation>
                            <xs:documentation>This is the delta from the scheduled
departure date time. </xs:documentation>
                        </xs:annotation>
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="Reason" type="cool:DescriptionT"
minOccurs="0"/>
                                <xs:element name="Time" type="cool:DateTimeT"
minOccurs="0"/>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element name="EstimatedTime" minOccurs="0">
            <xs:annotation>
                <xs:documentation>This is the estimated time of carrier arrival or
departure (when a carrier is delayed).</xs:documentation>
            </xs:annotation>
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="Location" type="DateTimeLocT" minOccurs="0"/>
                    <xs:element name="DateTime" type="DateTimeT"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="MovePlanInfoT">
    <xs:annotation>
        <xs:documentation>Information about a movement plan</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="Desc" minOccurs="0">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="4000"/>
                    <xs:whiteSpace value="preserve"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="Mission" minOccurs="0">
            <xs:annotation>

```

```

        <xs:documentation>The mission statement</xs:documentation>
      </xs:annotation>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="4000"/>
        <xs:whiteSpace value="preserve"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="ConOps" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="4000"/>
        <xs:minLength value="1"/>
        <xs:whiteSpace value="preserve"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="Assumptions" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="4000"/>
        <xs:minLength value="1"/>
        <xs:whiteSpace value="preserve"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="CommencementDate" type="xs:date" minOccurs="0">
    <xs:annotation>
      <xs:documentation>Calendar date that plan is executed. All planned
dates are relative to this date</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="PlanState" minOccurs="0">
    <xs:annotation>
      <xs:documentation>DRAFT, APPROVED, BASELINED or
ARCHIVED</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="DRAFT"/>
        <xs:enumeration value="APPROVED"/>
        <xs:enumeration value="BASELINED"/>
        <xs:enumeration value="ARCHIVED"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
</xs:sequence>
<xs:attribute name="planId" type="IdT" use="required"/>
<xs:attribute name="version" use="required">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="5"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
</xs:complexType>
<xs:complexType name="RequestToMoveT">
  <xs:annotation>
    <xs:documentation>Describes a request to move. This is not associated with
any plan; REQID identifies the specific request</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="RequestDate" type="DateTimeT" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The date the user submitted the request for lift
support (ie the data this instance documanet was created.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Reason" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The reason for the request</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="4000"/>
          <xs:whiteSpace value="preserve"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>

```

```

        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="State">
      <xs:annotation>
        <xs:documentation>PENDING - request has been submitted, but may not be
complete; READY - request has all fields necessary for feasibility filled in; APPROVED
- the request is now a requirement; REJECTED - request needs to be modified by the
submitter</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="APPROVED"/>
          <xs:enumeration value="PENDING"/>
          <xs:enumeration value="REJECTED"/>
          <xs:enumeration value="READY"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Organization" type="cool:OrgInfoT" minOccurs="0"
maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>List of organizations or sub-organization (possibly
down to individual persons) that need to be moved</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Equipment" type="EquipShipT" minOccurs="0"
maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>This is the list of non-containerized equipment that
is planned to be moved. </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Supply" type="SupplyItemT" minOccurs="0"
maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>This is a list of supplies</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Container" type="cool:ContainerT" minOccurs="0"
maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>List of containerized equipment, empty container(s)
needing redeployment</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="PersonList" type="cool:PersonT" minOccurs="0"
maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>List of people being moved</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="PAX" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Number of personnel being moved. ORGID is the
requesting organization (NEOs) otherwise owning organization</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:integer">
            <xs:attribute name="orgId" type="IdT" use="required"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="ArrivalLoc" type="cool:LocationT" minOccurs="0"/>
    <xs:element name="DepartLoc" type="cool:LocationT" minOccurs="0"/>
    <xs:element name="PrefTransportMode" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Transportation Mode</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="AIR"/>
          <xs:enumeration value="LAND_RAIL"/>
          <xs:enumeration value="LAND_MOTOR"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

```

        <xs:enumeration value="SEA"/>
        <xs:enumeration value="ORGANIC_AIR"/>
        <xs:enumeration value="ORGANIC_LAND"/>
        <xs:enumeration value="ORGANIC_SEA"/>
        <xs:enumeration value="ANY"/>
    </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="PointOfContact" type="cool:PointOfContactT" minOccurs="0"/>
<xs:element name="Comments" type="cool:DescriptionT" minOccurs="0"/>
<xs:element name="AvailLoadDate" type="DateTimeT" minOccurs="0">
    <xs:annotation>
        <xs:documentation>Date requirement is at POE and can be
loaded</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="EarliestArrivalDate" type="DateTimeT" minOccurs="0">
    <xs:annotation>
        <xs:documentation>Earliest date the requirement should reach the
POD</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="LatestArrivalDate" type="DateTimeT" minOccurs="0">
    <xs:annotation>
        <xs:documentation>Latest date the requirement should reach the
POD</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
<xs:attribute name="requestId" type="IdT" use="required"/>
<xs:attribute name="requestVersion" type="cool:VersionT" use="required"/>
</xs:complexType>
<xs:complexType name="MoveReqT">
    <xs:annotation>
        <xs:documentation>ID uniquely identifies a movement requirement in the
coalition domain; MOVEREQID uniquely identifies a movement requirement in the
coalition domain; MOVEREQVERSION specifies the version of the movement
requirement</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="Organization" type="cool:OrgInfoT" minOccurs="0"
maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>List of organizations or sub-organization (possibly
down to individual persons) that need to be moved</xs:documentation>
            </xs:annotation>
</xs:element>
        <xs:element name="Equipment" type="EquipShipT" minOccurs="0"
maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>This is the list of non-containerized equipment that
is planned to be moved. </xs:documentation>
            </xs:annotation>
</xs:element>
        <xs:element name="Supply" type="SupplyItemT" minOccurs="0"
maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>This is a list of supplies</xs:documentation>
            </xs:annotation>
</xs:element>
        <xs:element name="Container" type="ContainerT" minOccurs="0"
maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>List of containerized equipment, empty container(s)
needing redeployment</xs:documentation>
            </xs:annotation>
</xs:element>
        <xs:element name="Person" type="cool:PersonT" minOccurs="0"
maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>List of people being moved</xs:documentation>
            </xs:annotation>
</xs:element>
        <xs:element name="PAX" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Number of personnel being moved. ORGID is the
requesting organization (NEOs) otherwise owning organization</xs:documentation>
            </xs:annotation>

```

```

        </xs:annotation>
        <xs:complexType>
            <xs:simpleContent>
                <xs:extension base="xs:integer">
                    <xs:attribute name="orgId" type="IdT" use="required"/>
                </xs:extension>
            </xs:simpleContent>
        </xs:complexType>
    </xs:element>
    <xs:element name="Arrival" type="cool:DateTimeLocT">
        <xs:annotation>
            <xs:documentation>Arrival location, date and time
(hour)</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="Departure" type="cool:DateTimeLocT">
        <xs:annotation>
            <xs:documentation>Departure location, date and time
(hour)</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="TransportMode">
        <xs:annotation>
            <xs:documentation>Mode of transportation</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="AIR"/>
                <xs:enumeration value="LAND_RAIL"/>
                <xs:enumeration value="LAND_MOTOR"/>
                <xs:enumeration value="SEA"/>
                <xs:enumeration value="ORGANIC_AIR"/>
                <xs:enumeration value="ORGANIC_LAND"/>
                <xs:enumeration value="ORGANIC_SEA"/>
                <xs:enumeration value="ANY"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="ReqDeliveryDate" type="cool:DateTimeT" minOccurs="0">
        <xs:annotation>
            <xs:documentation>The date the requirement must be at the
POD</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="CoalitionReqDate" type="cool:DateTimeT" minOccurs="0">
        <xs:annotation>
            <xs:documentation>The date the requirement must be mission ready or
the mission will be impacted </xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="EarliestArrivalDate" type="cool:DateTimeT" minOccurs="0">
        <xs:annotation>
            <xs:documentation>The earliest the requirement should arrive at the
POD</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="LatestArrivalDate" type="cool:DateTimeT" minOccurs="0">
        <xs:annotation>
            <xs:documentation>The latest the requirement should arrive at the
POD</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="AvailLoadDate" type="cool:DateTimeT" minOccurs="0">
        <xs:annotation>
            <xs:documentation>The date the requirement is available to be loaded
at the POE</xs:documentation>
        </xs:annotation>
    </xs:element>
</xs:sequence>
<xs:attribute name="moveReqId" type="IdT" use="required"/>
<xs:attribute name="moveReqVersion" type="cool:VersionT" use="required"/>
</xs:complexType>
<xs:complexType name="MovePlanSectT">
    <xs:annotation>

```

```

        <xs:documentation>SECTIONID uniquely identified a plan section in the
        coalition domain. There can be one or more sections for a given
        plan.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="Desc" minOccurs="0">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:minLength value="0"/>
                    <xs:maxLength value="4000"/>
                    <xs:whiteSpace value="preserve"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="MoveReq" type="cool:MoveReqT" maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>MOVEREQID uniquely identifies a movement requirement
                in the coalition domain; PLANID uniquely identifies a movement plan in the coalition
                domain</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="sectId" type="IdT" use="required"/>
</xs:complexType>
<xs:complexType name="MovementPlanT">
    <xs:annotation>
        <xs:documentation>ID movement plan Id; VERSION specifies the version of this
        plan</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="cool:MovePlanInfoT">
            <xs:sequence>
                <xs:element name="MovPlanSec" type="cool:MovePlanSectT"
                maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="LengthT">
    <xs:annotation>
        <xs:documentation>MEASSYS defines the measurement system; UNIT the unit of
        measure</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute name="measSys" use="required">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="METRIC"/>
                        <xs:enumeration value="ENGLISH"/>
                        <xs:enumeration value="OTHER"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="lengthUnit" use="required">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="NAUTICAL_MILE"/>
                        <xs:enumeration value="MILE"/>
                        <xs:enumeration value="YARD"/>
                        <xs:enumeration value="FEET"/>
                        <xs:enumeration value="INCH"/>
                        <xs:enumeration value="FATHOM"/>
                        <xs:enumeration value="KILOMETER"/>
                        <xs:enumeration value="METER"/>
                        <xs:enumeration value="CENTIMETER"/>
                        <xs:enumeration value="MILIMETER"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="VolumeT">
    <xs:annotation>
        <xs:documentation>MEASSYS defines the measurement system; UNIT the unit of
        measure</xs:documentation>

```

```

</xs:annotation>
<xs:simpleContent>
  <xs:extension base="xs:string">
    <xs:attribute name="measSys" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="ENGLISH"/>
          <xs:enumeration value="METRIC"/>
          <xs:enumeration value="OTHER"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="volumeUnit" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="GALLON"/>
          <xs:enumeration value="PINT"/>
          <xs:enumeration value="FLUID_OUNCE"/>
          <xs:enumeration value="LITER"/>
          <xs:enumeration value="METER_CUBED"/>
          <xs:enumeration value="MTON"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="WeightT">
  <xs:annotation>
    <xs:documentation>MEASSYS defines the measurement system; UNIT the unit of
measure</xs:documentation>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="measSys" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="METRIC"/>
            <xs:enumeration value="ENGLISH"/>
            <xs:enumeration value="OTHER"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="weightUnit" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="LONG_TON"/>
            <xs:enumeration value="SHORT_TON"/>
            <xs:enumeration value="TON"/>
            <xs:enumeration value="POUND"/>
            <xs:enumeration value="OUNCE"/>
            <xs:enumeration value="KILOGRAM"/>
            <xs:enumeration value="GRAM"/>
            <xs:enumeration value="MILIGRAM"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:complexType name="AreaT">
  <xs:annotation>
    <xs:documentation>MEASSYS defines the measurement system; UNIT the unit of
measure</xs:documentation>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="measSys" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="ENGLISH"/>
            <xs:enumeration value="METRIC"/>
            <xs:enumeration value="OTHER"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

```

        <xs:attribute name="areaUnit" use="required">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="ACRE"/>
              <xs:enumeration value="SQUARE_MILE"/>
              <xs:enumeration value="METER_SQUARED"/>
              <xs:enumeration value="KILOMETER_SQUARED"/>
              <xs:enumeration value="YARD_SQUARED"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="BaseSupplyItemT">
    <xs:annotation>
      <xs:documentation>Describes a generic supply item not associated with an
organization. SUPPLYID uniquely identifies a type of supply item in the coalition
domain</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="Name" type="cool:NameT" minOccurs="0">
        <xs:annotation>
          <xs:documentation>The NOMEN or clear text name of the supply item.
</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Code">
        <xs:annotation>
          <xs:documentation>The specific catalog coding system. For example this
could be a NSN, a UPC, a PART NUMBER, etc. CODETYPE specifies the kind of
code.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="BaseNameT">
              <xs:attribute name="codeType" use="required">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:maxLength value="100"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:attribute>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="UnitOfIssue" minOccurs="0">
        <xs:annotation>
          <xs:documentation>The way this item is packaged.</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="AMPOULE"/>
            <xs:enumeration value="ASSEMBLY"/>
            <xs:enumeration value="ASSORTMENT"/>
            <xs:enumeration value="BAG"/>
            <xs:enumeration value="BALE"/>
            <xs:enumeration value="BALL"/>
            <xs:enumeration value="BAR"/>
            <xs:enumeration value="BARREL"/>
            <xs:enumeration value="BOARD_FOOT"/>
            <xs:enumeration value="BOLT"/>
            <xs:enumeration value="BOOK"/>
            <xs:enumeration value="BOTTLE"/>
            <xs:enumeration value="BOX"/>
            <xs:enumeration value="BUNDLE"/>
            <xs:enumeration value="CAKE"/>
            <xs:enumeration value="CAN"/>
            <xs:enumeration value="CARBOY"/>
            <xs:enumeration value="CARTRIDGE"/>
            <xs:enumeration value="COIL"/>
            <xs:enumeration value="CONE"/>
            <xs:enumeration value="CONTAINER"/>
            <xs:enumeration value="CUBIC_METER"/>
            <xs:enumeration value="CUBIC_FOOT"/>
            <xs:enumeration value="CUBIC_YARD"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

```

```

        <xs:enumeration value="CYLINDER"/>
        <xs:enumeration value="DOZEN"/>
        <xs:enumeration value="DRUM"/>
        <xs:enumeration value="EACH"/>
        <xs:enumeration value="FIFTY"/>
        <xs:enumeration value="FIVE"/>
        <xs:enumeration value="FOOT"/>
        <xs:enumeration value="GALLON"/>
        <xs:enumeration value="GROSS"/>
        <xs:enumeration value="GROUP"/>
        <xs:enumeration value="HANK"/>
        <xs:enumeration value="HUNDRED"/>
        <xs:enumeration value="INCH"/>
        <xs:enumeration value="JAR"/>
        <xs:enumeration value="KIT"/>
        <xs:enumeration value="LENGTH"/>
        <xs:enumeration value="LITER"/>
        <xs:enumeration value="MEAL"/>
        <xs:enumeration value="METER"/>
        <xs:enumeration value="OUNCE"/>
        <xs:enumeration value="OUTFIT"/>
        <xs:enumeration value="PACKAGE"/>
        <xs:enumeration value="PACKET"/>
        <xs:enumeration value="PAD"/>
        <xs:enumeration value="PAGE"/>
        <xs:enumeration value="PAIR"/>
        <xs:enumeration value="PINT"/>
        <xs:enumeration value="PLATE"/>
        <xs:enumeration value="POUND"/>
        <xs:enumeration value="QUART"/>
        <xs:enumeration value="RATION"/>
        <xs:enumeration value="REAM"/>
        <xs:enumeration value="REEL"/>
        <xs:enumeration value="ROLL"/>
        <xs:enumeration value="SET"/>
        <xs:enumeration value="SHEET"/>
        <xs:enumeration value="SHOT"/>
        <xs:enumeration value="SKEIN"/>
        <xs:enumeration value="SKID"/>
        <xs:enumeration value="SPOOL"/>
        <xs:enumeration value="SQUARE_FOOT"/>
        <xs:enumeration value="SQUARE_YARD"/>
        <xs:enumeration value="STICK"/>
        <xs:enumeration value="STRIP"/>
        <xs:enumeration value="TEN"/>
        <xs:enumeration value="THIRTY_SIX"/>
        <xs:enumeration value="THOUSAND"/>
        <xs:enumeration value="THOUSAND_CUBIC_FOOT"/>
        <xs:enumeration value="TON"/>
        <xs:enumeration value="TROY_OUNCE"/>
        <xs:enumeration value="TUBE"/>
        <xs:enumeration value="TWENTY_FIVE"/>
        <xs:enumeration value="TWENTY_FOUR"/>
        <xs:enumeration value="VIAL"/>
        <xs:enumeration value="YARD"/>
    </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="Quantity">
    <xs:annotation>
        <xs:documentation>The number of these items</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:integer">
            <xs:minInclusive value="0"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="Desc" minOccurs="0">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="4000"/>
            <xs:whiteSpace value="preserve"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>

```

```

<xs:element name="Length" type="cool:LengthT"/>
<xs:element name="Height" type="cool:LengthT"/>
<xs:element name="Width" type="cool:LengthT"/>
<xs:element name="Volume" type="cool:VolumeT" minOccurs="0"/>
<xs:element name="Weight" type="cool:WeightT"/>
<xs:element name="HazCargo" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Hazardous Cargo code.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="BaseNameT">
        <xs:attribute name="codeType" type="BaseNameT" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="NetExplosiveQuantity" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="100"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
</xs:sequence>
<xs:attribute name="supplyId" type="IdT" use="required"/>
</xs:complexType>
<xs:complexType name="SupplyItemT">
  <xs:annotation>
    <xs:documentation>SUPPLYID uniquely identifies a type of supply item in the
coalition domain</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="BaseSupplyItemT">
      <xs:sequence>
        <xs:element name="ProvidingOrgId" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="SupplyTransT">
  <xs:annotation>
    <xs:documentation>A list of supply items that are being
transported.</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="SupplyItemT">
      <xs:sequence>
        <xs:element name="Congisnor" type="ConsignorT" minOccurs="0"/>
        <xs:element name="Consignee" type="ConsignorT" minOccurs="0"/>
        <xs:element ref="TransitNum"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="AddressT">
  <xs:annotation>
    <xs:documentation>Defines the basica postal adres structure to be used in
the coalition</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Street1" type="xs:string" minOccurs="0"/>
    <xs:element name="Street2" type="xs:string" minOccurs="0"/>
    <xs:element name="Street3" type="xs:string" minOccurs="0"/>
    <xs:element name="Street4" type="xs:string" minOccurs="0"/>
    <xs:element name="County" type="xs:string" minOccurs="0"/>
    <xs:element name="State" type="xs:string" minOccurs="0"/>
    <xs:element name="PostalCode" type="xs:string" minOccurs="0"/>
    <xs:element name="Country" type="xs:string" minOccurs="0"/>
    <xs:element name="RefCode" minOccurs="0"/>
  </xs:sequence>
  <xs:annotation>
    <xs:documentation>This could be like a DODAAC, or some other nation
specific addressing scheme.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">

```

```

        <xs:attribute name="codeType" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="ConsignorT">
  <xs:annotation>
    <xs:documentation>Defines a consignor or consignee. ADDRID (optional)
uniquely identifies this consignor or consignee. </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="AddressT">
      <xs:sequence minOccurs="0">
        <xs:element name="Name" type="NameT" minOccurs="0"/>
        <xs:element name="PointOfContact" type="cool:PointOfContactT"
minOccurs="0"/>
        <xs:element name="OrgId" type="xs:string" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Identifies the organization this consignor or
consignee is associated with.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="Comments" minOccurs="0">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="4000"/>
              <xs:whiteSpace value="preserve"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="addrId" type="IdT" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="CargoITVT">
  <xs:annotation>
    <xs:documentation>Defines a raw cargo ITV record. TIMESTAMP is the date time
the record was created by the ITV system. FORMAT is the format of the
timestamp.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Timestamp" type="DateTimeT"/>
    <xs:element name="Consignor" type="cool:ConsignorT" minOccurs="0"/>
    <xs:element name="Consignee" type="cool:ConsignorT" minOccurs="0"/>
    <xs:element name="CarrierId" minOccurs="0">
      <xs:annotation>
        <xs:documentation>This Id uniquely identifies a carrier in the
coalition domain. </xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="CarrierT"/>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="CarrierStatus" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="ONSTATION"/>
          <xs:enumeration value="ARRIVING"/>
          <xs:enumeration value="DEPARTING"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="MissionNumber" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Mission Number</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="numberType"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>

```

```

        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="VoyageNumber" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Voyage Document Number</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string"/>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="CommodityCode" type="xs:string" minOccurs="0"/>
    <xs:element name="BillOfLading" minOccurs="0">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="billType" use="required">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="GOVERNMENT"/>
                  <xs:enumeration value="COMMERCIAL"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="POE" type="cool:LocationT" minOccurs="0"/>
    <xs:element name="CurrentLocation" minOccurs="0">
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="cool:LocationT">
            <xs:sequence>
              <xs:element name="Arrival" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="ScheduledDateTime"
type="cool:DateTimeT" minOccurs="0">
                      <xs:annotation>
                        <xs:documentation>Scheduled location, date
and time.</xs:documentation>
                      </xs:annotation>
                    </xs:element>
                    <xs:element name="Actual" minOccurs="0">
                      <xs:annotation>
                        <xs:documentation>Actual location date and
time.</xs:documentation>
                      </xs:annotation>
                    </xs:complexType>
                    <xs:sequence minOccurs="0">
                      <xs:element name="DateTime"
type="cool:DateTimeT"/>
                      <xs:element name="Delay" minOccurs="0">
                        <xs:annotation>
                          <xs:documentation>This is the delta
from the scheduled departure date time. </xs:documentation>
                        </xs:annotation>
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="Reason"
type="cool:DescriptionT" minOccurs="0"/>
                          <xs:element name="Time"
type="cool:DateTimeT" minOccurs="0"/>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
                <xs:element name="EstimatedDateTime"
type="cool:DateTimeT" minOccurs="0">
                  <xs:annotation>
                    <xs:documentation>This is the estimated time
of carrier arrival when a carrier is delayed.</xs:documentation>
                  </xs:annotation>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>

```

```

        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Departure" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ScheduledDateTime"
type="cool:DateTimeT" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Scheduled location, date
and time.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Actual" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Actual location date and
time.</xs:documentation>
        </xs:annotation>
      </xs:complexType>
      <xs:sequence minOccurs="0">
        <xs:element name="DateTime"
type="cool:DateTimeT"/>
        <xs:element name="Delay" minOccurs="0">
          <xs:annotation>
            <xs:documentation>This is the delta
from the scheduled departure date time. </xs:documentation>
          </xs:annotation>
        </xs:complexType>
        <xs:sequence>
          <xs:element name="Reason"
type="cool:DescriptionT" minOccurs="0"/>
          <xs:element name="Time"
type="cool:DateTimeT" minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="EstimatedDateTime"
type="cool:DateTimeT" minOccurs="0">
  <xs:annotation>
    <xs:documentation>This is the estimated time
of carrier departure when a carrier is delayed.</xs:documentation>
  </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="locationType" use="required">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="ARRIVE"/>
      <xs:enumeration value="DEPART"/>
      <xs:enumeration value="ONHAND"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:element>
<xs:element name="POD" type="cool:LocationT" minOccurs="0"/>
<xs:element name="TotalNumberOfPieces" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="TotalWeight" type="cool:WeightT" minOccurs="0"/>
<xs:element name="Organization" type="OrgTransT" minOccurs="0"
maxOccurs="unbounded"/>

```

```

<xs:element name="Person" minOccurs="0" maxOccurs="unbounded">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="PersonTransT"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:element name="Equipment" minOccurs="0" maxOccurs="unbounded">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="cool:EquipTransT"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:element name="Supply" type="SupplyTransT" minOccurs="0"
maxOccurs="unbounded"/>
<xs:element name="Container" type="cool:ContainerTransT" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="cargoITVId" type="IdT" use="optional"/>
</xs:complexType>
<xs:complexType name="PointOfContactT">
  <xs:annotation>
    <xs:documentation>Defines an person name, address, phone number and email
address</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="cool:PersonNmT">
      <xs:sequence>
        <xs:element name="PhoneNum1" type="cool:PhoneNumT" minOccurs="0"/>
        <xs:element name="PhoneNum2" type="cool:PhoneNumT" minOccurs="0"/>
        <xs:element name="FaxNum" type="cool:PhoneNumT" minOccurs="0"/>
        <xs:element name="EmailAddress" minOccurs="0">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="100"/>
              <xs:whiteSpace value="collapse"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="WebSiteURL" type="xs:string" minOccurs="0"/>
        <xs:element name="OrgId" type="xs:string" minOccurs="0"/>
        <xs:element name="Address" type="AddressT" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="personId" type="IdT" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="TransitIdT">
  <xs:annotation>
    <xs:documentation>Provides a list of translated transit ids. TRANSTID is a
coalition tracking number that is mapped to one or more nation-specific tracking
number(s); </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Comments" type="cool:DescriptionT" minOccurs="0"/>
    <xs:element name="National">
      <xs:annotation>
        <xs:documentation>Organization (Org_Id) that owns this tracking
number.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="TrackingNum" maxOccurs="unbounded">
            <xs:annotation>
              <xs:documentation>A list of one or more nation specific
tracking numbers that are associated with a given Transit Id. ORGID specifies the
owning organization; TRACKNUMTYPE defines the type of tracking number.
</xs:documentation>
            </xs:annotation>
            <xs:complexType>
              <xs:simpleContent>
                <xs:extension base="cool:IdT">
                  <xs:attribute name="orgId" type="cool:IdT"
use="optional"/>
                  <xs:attribute name="trackNumType" type="cool:IdT"
use="required"/>
                </xs:extension>
              </xs:simpleContent>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>

```

```

        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="Comments" type="cool:DescriptionT"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="transitId" type="IdT" use="required"/>
</xs:complexType>
<xs:complexType name="TransitIdListT">
  <xs:annotation>
    <xs:documentation>A list of transit Ids</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="TransitId" type="TransitIdT" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="PhoneNumT">
  <xs:annotation>
    <xs:documentation>FORMAT; TYPE</xs:documentation>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="format" type="xs:string" use="optional"/>
      <xs:attribute name="type" type="xs:string" use="optional"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:complexType name="CapabilityInfoT">
  <xs:annotation>
    <xs:documentation>Description of unit capabilities.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="CapabilityTypeCd">
      <xs:annotation>
        <xs:documentation>Type of a capability (Transportation, Services,
Engineering, Medical, Supplies, Maintainance, etc.)</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="19"/>
          <xs:enumeration value="ENGINEERING"/>
          <xs:enumeration value="MEDICAL"/>
          <xs:enumeration value="MAINTENANCE"/>
          <xs:enumeration value="SERVICES"/>
          <xs:enumeration value="SUPPLY"/>
          <xs:enumeration value="TRANSPORTATION"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="ShortTitle">
      <xs:annotation>
        <xs:documentation>Short title for a capability</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="12"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="LongTitle">
      <xs:annotation>
        <xs:documentation>Long title for a capability</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="30"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="CompleteTitle">
      <xs:annotation>
        <xs:documentation>Complete title for a capability</xs:documentation>

```

```

</xs:annotation>
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:maxLength value="65"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="Descr" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="2000"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="FunctionalArea">
  <xs:annotation>
    <xs:documentation>Functional area for this capability (Health
Services, Transportation, Supply Systems, General Engineering, etc)</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="19"/>
      <xs:enumeration value="General Engineering"/>
      <xs:enumeration value="Health Services"/>
      <xs:enumeration value="Maintenance"/>
      <xs:enumeration value="Other Services"/>
      <xs:enumeration value="Supply Systems"/>
      <xs:enumeration value="Transportation"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element ref="UnitOfMeasure"/>
<xs:element name="QuantQualInd" type="xs:boolean">
  <xs:annotation>
    <xs:documentation>Qualitative or quantitative
indication.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="QuantValue" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Quantitative values</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:float"/>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="QualValue" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Qualitative values</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Echelon 2"/>
      <xs:enumeration value="Echelon 3"/>
      <xs:enumeration value="Echelon 4"/>
      <xs:enumeration value="Echelon 5"/>
      <xs:enumeration value="INTERMEDIATE"/>
      <xs:enumeration value="LIMITED"/>
      <xs:enumeration value="SIGNIFICANT"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Uplddt" type="xs:dateTime">
  <xs:annotation>
    <xs:documentation>System date time the record uploaded into the
system.</xs:documentation>
    <xs:documentation>Date the record uploaded into the
system.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="Datasrc">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="6"/>
    </xs:restriction>
  </xs:simpleType>

```

```

        </xs:simpleType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="capabilityId" use="required">
      <xs:annotation>
        <xs:documentation>An identification of a capability.</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="10"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
  <xs:complexType name="UICInforT">
    <xs:annotation>
      <xs:documentation>Describes an unit (UIC)</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="AbbrevNM" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="45"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="UnitLastUpDt" type="xs:date" minOccurs="0"/>
      <xs:element name="ParentUIC" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="6"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="LocLastUpDt" type="xs:date" minOccurs="0"/>
      <xs:element name="ClassLevelCd" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Classified level for this UIC (Unclassified or
Classified)</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Component" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Unit service component.</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="15"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Status" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="7"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="HomeStation" minOccurs="0">
        <xs:annotation>
          <xs:documentation>A location code. It could be GEOLoc code, IATA
code, ICAO code, etc</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="38"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="HomeStationCdType" minOccurs="0">
        <xs:annotation>

```

```

        <xs:documentation>Type of location code (GEOLoc, IATA, ICAO,
etc.)</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="18"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="PlanId" minOccurs="0">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="5"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="DataSrc">
    <xs:annotation>
        <xs:documentation>Data source for this UIC.</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="8"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="Uplddt" type="xs:dateTime">
    <xs:annotation>
        <xs:documentation>System date time this UIC is loaded into the
system.</xs:documentation>
    </xs:annotation>
</xs:element>
<!-- <xs:element name="RatingLastChngDt" minOccurs="0"/>
<xs:element name="RatingLastUpDt" minOccurs="0"/>
<xs:element name="OverallRating" minOccurs="0"/>
<xs:element name="SplyRating" minOccurs="0"/>
<xs:element name="SplyCd" minOccurs="0"/>
<xs:element name="EquipRating" minOccurs="0"/>
<xs:element name="EquipCd" minOccurs="0"/>
<xs:element name="TrngRating" minOccurs="0"/>
<xs:element name="TrngCd" minOccurs="0"/>
<xs:element name="PrslRating" minOccurs="0"/>
<xs:element name="PrslCd" minOccurs="0"/> -->
</xs:sequence>
<xs:attribute name="UIC" use="required">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="6"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
</xs:complexType>
<xs:complexType name="UnitTypeInfoT">
    <xs:annotation>
        <xs:documentation>Description of an Unit Type (UTC)</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element ref="UnitCode" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="UnitTypeNM" minOccurs="0">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="60"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="ServiceCd" minOccurs="0">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="1"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="WartimePersonQty" type="xs:integer" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Number of personnel required to execute
UTC.</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>

```

```

        </xs:element>
        <xs:element name="NonOrgTransPRSL" type="xs:integer" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Number of Non Organic Transportation
personnel</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="UTCCreateDt" type="xs:date" minOccurs="0"/>
        <xs:element name="UTCLastUpDt" type="xs:date" minOccurs="0"/>
        <xs:element name="ParentUTC" minOccurs="0">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="5"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="EchelonCd" minOccurs="0">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="3"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="MissionDesc" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Mission desription</xs:documentation>
            </xs:annotation>
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="50"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="SRC9" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Source Requirement Code. The U.S. Army code based
upon the Tables of Equipment and Organization (TOEs), and identifying a particular
Type Unit or Actual Unit.</xs:documentation>
            </xs:annotation>
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="9"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="S2525AHierarchyCd" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Symbology hierarchy code</xs:documentation>
            </xs:annotation>
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="20"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="S2525AAffiliationCd" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Symbology affiliation code</xs:documentation>
            </xs:annotation>
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="3"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="S2525AEchelonCd" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Symbology echelon code</xs:documentation>
            </xs:annotation>
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="3"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="S2525AIndicators" minOccurs="0">

```

```

        <xs:annotation>
          <xs:documentation>Symbology indicator</xs:documentation>
        </xs:annotation>
      </xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="3"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="UnitRoleCd" minOccurs="0">
    <xs:annotation>
      <xs:documentation>Unit role code. For example, for Unit Role
Headquarters unit role code would be C, and J is for Supply, etc.</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="1"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="PlanId" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="5"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="UpldDt" type="xs:dateTime">
    <xs:annotation>
      <xs:documentation>System date that this record is loaded into the
system</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="DataSrc">
    <xs:annotation>
      <xs:documentation>Data source of this record.</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="8"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
</xs:sequence>
<xs:attribute name="UTC" use="required">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="5"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
</xs:complexType>
<xs:complexType name="OrgTransT">
  <xs:complexContent>
    <xs:extension base="OrgInfoT">
      <xs:sequence>
        <xs:element ref="TransitNum"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="TransitNum" type="cool:TransitIdT">
  <xs:annotation>
    <xs:documentation>Generic coalition transit id. This transit Id can have one
or more national tracking numbers.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="UnitTypeCapabilityT">
  <xs:annotation>
    <xs:documentation>Describes capability for an unit type</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="UnitTypeInfoT">
      <xs:sequence>
        <xs:element ref="Capability" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>

```

```

        </xs:complexContent>
    </xs:complexType>
    <xs:element name="UnitCode" type="UICInforT">
        <xs:annotation>
            <xs:documentation>Description of an UIC</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="Capability" type="CapabilityInfoT">
        <xs:annotation>
            <xs:documentation>Description of logistics capability</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="UnitOfMeasure">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="30"/>
                <xs:enumeration value="Ambulances"/>
                <xs:enumeration value="Ambulat/Cycle"/>
                <xs:enumeration value="Beds"/>
                <xs:enumeration value="Casualties/Hr"/>
                <xs:enumeration value="Cu Ft"/>
                <xs:enumeration value="Echelon 2,Echelon 3,Echelon 4"/>
                <xs:enumeration value="Echelon 3,Echelon 4,Echelon 5"/>
                <xs:enumeration value="GPM"/>
                <xs:enumeration value="Gal"/>
                <xs:enumeration value="Gal/Hr"/>
                <xs:enumeration value="Hrs/Day"/>
                <xs:enumeration value="Intermediate Level,depot Level"/>
                <xs:enumeration value="LIMITED,SIGNIFICANT"/>
                <xs:enumeration value="Lbs/Hr"/>
                <xs:enumeration value="Litter/Cycle"/>
                <xs:enumeration value="MT"/>
                <xs:enumeration value="Meals/Day"/>
                <xs:enumeration value="Oper. Rooms"/>
                <xs:enumeration value="PAX"/>
                <xs:enumeration value="Patients"/>
                <xs:enumeration value="Patients/Cycle"/>
                <xs:enumeration value="Patients/Hr"/>
                <xs:enumeration value="Persons/Hr"/>
                <xs:enumeration value="Providers"/>
                <xs:enumeration value="Remains/Day"/>
                <xs:enumeration value="ST"/>
                <xs:enumeration value="ST/Cycle"/>
                <xs:enumeration value="ST/Day"/>
                <xs:enumeration value="Ships"/>
                <xs:enumeration value="Units/Day"/>
                <xs:enumeration value="Warrant Amt"/>
                <xs:enumeration value="kw"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:complexType name="UnitCapResult">
        <xs:annotation>
            <xs:documentation>Result instance for Unit Capability</xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="CapResult" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:simpleContent>
                        <xs:extension base="xs:string">
                            <xs:attribute name="capabilityId" type="xs:string"
use="required"/>
                        </xs:extension>
                    </xs:simpleContent>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="UTC" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="5"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>
    <xs:complexType name="UnitTypeResult">

```

```

<xs:annotation>
  <xs:documentation>Result instance for UTC</xs:documentation>
</xs:annotation>
<xs:simpleContent>
  <xs:extension base="xs:string">
    <xs:attribute name="UTC" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="5"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="HighLevelMovementPlanT">
  <xs:annotation>
    <xs:documentation>Hight level description of a movement
plan.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="PlanState" type="xs:string" minOccurs="0"/>
    <xs:element name="Mission" type="xs:string" minOccurs="0"/>
    <xs:element name="CommencementDate" type="xs:date" minOccurs="0"/>
    <xs:element name="Movement">
      <xs:annotation>
        <xs:documentation>MoveReqId uniquely identifies a movement requirement
in the coalition domain</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="State" type="xs:string" minOccurs="0"/>
          <xs:element name="Pax" minOccurs="0">
            <xs:annotation>
              <xs:documentation>Number of Personnel for this movement
plan. OrgId is the owning organization.</xs:documentation>
            </xs:annotation>
            <xs:complexType>
              <xs:simpleContent>
                <xs:extension base="xs:string">
                  <xs:attribute name="orgId" type="IdT" use="required"/>
                </xs:extension>
              </xs:simpleContent>
            </xs:complexType>
          </xs:element>
          <xs:element name="Organization" minOccurs="0"
maxOccurs="unbounded">
            <xs:annotation>
              <xs:documentation>List of organizations</xs:documentation>
            </xs:annotation>
            <xs:complexType>
              <xs:simpleContent>
                <xs:extension base="xs:string">
                  <xs:attribute name="orgId" type="IdT" use="required"/>
                </xs:extension>
              </xs:simpleContent>
            </xs:complexType>
          </xs:element>
          <xs:element name="Equipment" minOccurs="0" maxOccurs="unbounded">
            <xs:annotation>
              <xs:documentation>List of equipment items
.</xs:documentation>
            </xs:annotation>
            <xs:complexType>
              <xs:simpleContent>
                <xs:extension base="xs:string">
                  <xs:attribute name="equipId" type="IdT"
use="required"/>
                  <xs:attribute name="owningOrgId" type="IdT"
use="required"/>
                </xs:extension>
              </xs:simpleContent>
            </xs:complexType>
          </xs:element>
          <xs:element name="Container" minOccurs="0" maxOccurs="unbounded">
            <xs:annotation>
              <xs:documentation>List of containers.</xs:documentation>

```

```

        </xs:annotation>
        <xs:complexType>
            <xs:simpleContent>
                <xs:extension base="xs:string">
                    <xs:attribute name="containerId" type="IdT"
use="required"/>
                </xs:extension>
            </xs:simpleContent>
        </xs:complexType>
    </xs:element>
    <xs:element name="Supply" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
            <xs:documentation>List of supplies.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:simpleContent>
                <xs:extension base="xs:string">
                    <xs:attribute name="supplyId" type="IdT"
use="required"/>
                    <xs:attribute name="providingOrgId" type="IdT"
use="required"/>
                </xs:extension>
            </xs:simpleContent>
        </xs:complexType>
    </xs:element>
    <xs:element name="Person" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
            <xs:documentation>list of personnels.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:simpleContent>
                <xs:extension base="xs:string">
                    <xs:attribute name="personId" type="IdT"
use="required"/>
                </xs:extension>
            </xs:simpleContent>
        </xs:complexType>
    </xs:element>
    <xs:element name="Arrival" type="cool:DateTimeLocT">
        <xs:annotation>
            <xs:documentation>Arrival location, date, and
time.</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="Departure" type="cool:DateTimeLocT">
        <xs:annotation>
            <xs:documentation>Departure location, date, and
time.</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="TransportMode">
        <xs:annotation>
            <xs:documentation>Mode of transportation.</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="AIR"/>
                <xs:enumeration value="LAND_RAIL"/>
                <xs:enumeration value="LAND_MOTOR"/>
                <xs:enumeration value="SEA"/>
                <xs:enumeration value="ORGANIC_AIR"/>
                <xs:enumeration value="ORGANIC_LAND"/>
                <xs:enumeration value="ORGANIC_SEA"/>
                <xs:enumeration value="ANY"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="RequestDate" type="DateTimeT" minOccurs="0">
        <xs:annotation>
            <xs:documentation>Date requests for
movement</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="ReqDeliveryDate" type="DateTimeT" minOccurs="0">
        <xs:annotation>

```

```

        <xs:documentation>The date that requirement must be at the
POD</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="CoalitionReqDate" type="DateTimeT" minOccurs="0">
    <xs:annotation>
        <xs:documentation>The date the requirement must be ready for
mission.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="ApprovedDate" type="DateTimeT" minOccurs="0">
    <xs:annotation>
        <xs:documentation>Approved date for a
movement.</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
<xs:attribute name="moveReqId" type="cool:IdT" use="required"/>
<xs:attribute name="moveReqVersion" type="cool:VersionT"
use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="planId" type="IdT" use="required"/>
<xs:attribute name="version" type="xs:string" use="required"/>
<xs:attribute name="secId" type="IdT" use="optional"/>
</xs:complexType>
<xs:element name="CargoCode" type="cool:CargoCodeT">
    <xs:annotation>
        <xs:documentation>A coalition domain code that defines categories of
equipment. Based on US domain cargo codes.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:complexType name="CargoCodeT">
    <xs:annotation>
        <xs:documentation>A coalition domain code that defines categories of
equipment. Based on US domain cargo codes.</xs:documentation>
    </xs:annotation>
</xs:sequence>
<xs:element name="Category">
    <xs:annotation>
        <xs:documentation>Wheeled or tracked vehicles, non self-deploying
aircraft, floating craft, etc.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:simpleType>
    <xs:restriction base="xs:string">
        <xs:enumeration value="VEHICLES NON-SELF DEPLOYING"/>
        <xs:enumeration value="NON-SELF DEPLOYING AIRCRAFT"/>
        <xs:enumeration value="FLOATING CRAFT"/>
        <xs:enumeration value="HAZARD NON-VEHICLE"/>
        <xs:enumeration value="SCTY/HZD NON-VEHICLE"/>
        <xs:enumeration value="REFRIGERATED"/>
        <xs:enumeration value="BULK POL"/>
        <xs:enumeration value="BULK GRANULAR"/>
        <xs:enumeration value="OTHER NON-VEHICLE"/>
        <xs:enumeration value="SCTY/HZD VEHICLE"/>
        <xs:enumeration value="HAZARDOUS VEHICLE"/>
        <xs:enumeration value="AMMUNITION"/>
        <xs:enumeration value="NUCLEAR"/>
        <xs:enumeration value="CHEMICAL"/>
        <xs:enumeration value="VEHICLES SELF-DEPLOYING"/>
    </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="TransportDimension" type="xs:string" minOccurs="0">
    <xs:annotation>
        <xs:documentation>Describes the basic classes of equipment, cargo or
supply (i.e. non-air-transportable, outsized, oversized, bulk or organic
cargo)</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="Containerization" type="xs:string" minOccurs="0">
    <xs:annotation>
        <xs:documentation>Describes the categories of containers that this
equipment can be shipped in (i.e. 20 ft to 40 ft, not possible to put in a container
at all).</xs:documentation>
    </xs:annotation>
</xs:element>

```

```

        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:schema>

  <xs:schema targetNamespace="http://coalition/xml/schema"
    xmlns="http://coalition/xml/schema" xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:cool="http://coalition/xml/schema" elementFormDefault="qualified"
    attributeFormDefault="unqualified" version="1.0">
    <xs:include schemaLocation="./common/BaseTypes.xsd"/>
    <xs:element name="CargoITV">
      <xs:annotation>
        <xs:documentation>Defines a cargo ITV record</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Item" type="CargoITVT" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>

  <xs:schema targetNamespace="http://coalition/xml/schema"
    xmlns:cool="http://coalition/xml/schema" xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="http://coalition/xml/schema" elementFormDefault="qualified"
    attributeFormDefault="unqualified" version="1.0">
    <xs:include schemaLocation="./common/BaseTypes.xsd"/>
    <xs:element name="Carrier" type="cool:CarrierT">
      <xs:annotation>
        <xs:documentation>Comment describing your root element</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:schema>

  <xs:schema targetNamespace="http://coalition/xml/schema"
    xmlns="http://coalition/xml/schema" xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:cool="http://coalition/xml/schema" elementFormDefault="qualified"
    attributeFormDefault="unqualified" version="1.0">
    <xs:include schemaLocation="./common/BaseTypes.xsd"/>
    <xs:element name="CarrierSchedManifest">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Carrier" minOccurs="0" maxOccurs="unbounded">
            <xs:annotation>
              <xs:documentation>CARRIERID uniquely identifies a carrier in the
coalition domain.</xs:documentation>
            </xs:annotation>
            <xs:complexType>
              <xs:complexContent>
                <xs:extension base="cool:CarrierT">
                  <xs:sequence>
                    <xs:element name="ScheduleItem" minOccurs="0"
maxOccurs="unbounded">
                      <xs:annotation>
                        <xs:documentation>CarrierScheduleItemId uniquely
identifies a line in the carrier schedule; MISSIONVOYAGE could be a waybill number, a
commercial carrier tracking number, bill of lading number etc</xs:documentation>
                      </xs:annotation>
                      <xs:complexType>
                        <xs:complexContent>
                          <xs:extension base="cool:CarrierSchedT"/>
                        </xs:complexContent>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:extension>
              </xs:complexContent>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>

```

```

<xs:schema targetNamespace="http://coalition/xml/schema"
xmlns="http://coalition/xml/schema" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:cool="http://coalition/xml/schema" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="2.0">
  <xs:include schemaLocation="./common/BaseTypes.xsd"/>
  <xs:element name="EquipmentList">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Item" type="EquipTransT"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

<xs:schema targetNamespace="http://coalition/xml/schema"
xmlns="http://coalition/xml/schema" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:cool="http://coalition/xml/schema"
xmlns:ns1="http://coalition/xml/schema/movement" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="1.0">
  <xs:include schemaLocation="./common/BaseTypes.xsd"/>
  <xs:element name="MovementPlan" type="cool:MovementPlanT">
    <xs:annotation>
      <xs:documentation>Description of movement plans</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="MovementPlanList">
    <xs:annotation>
      <xs:documentation>List of high-level information of movement
plans</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="MovePlan" type="HighLevelMovementPlanT" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

<xs:schema targetNamespace="http://coalition/xml/schema"
xmlns="http://coalition/xml/schema" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:cool="http://coalition/xml/schema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:include schemaLocation="./common/BaseTypes.xsd"/>
  <xs:element name="MoveRequestList">
    <xs:annotation>
      <xs:documentation>List of Requests</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="cool:RequestToMove" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="RequestToMove" type="RequestToMoveT"/>
</xs:schema>

```


References

1. DSTO CTL SharePoint Portal Site, URL – <http://c2d-teams.dsto.defence.gov.au/task/04-072/CTL/default.aspx>
2. E. Kuster (January 2005) *Coalition Theatre Logistics (CTL) Advanced Concept Technology Demonstrator (ACTD) Architecture Overview*, DSTO-TN-0607, Defence Science and Technology Organisation, Command and Control Division
3. E. Kuster (in preparation) *Multi-lateral Information Exchange Environment (MIEE) Architecture*, Defence Science and Technology Organisation, Command and Control Division
4. P. Forster, E. Kuster (under review) *Coalition Theatre Logistics (CTL) Advanced Concept Technology Demonstrator (ACTD) Coalition Movement Requirement Plan Web Service Documentation*, Defence Science and Technology Organisation, Command and Control Division
5. Web Service-Interoperability Organization (WS-I), URL – <http://www.ws-i.org>
6. K. Ballinger, D. Ehnebuske, M. Gudgin, M. Nottingham, P. Yendluri (2004), *Basic Profile Version 1.0*, WS-I, URL – <http://www.ws-i.org/Profiles/BasicProfile-1.0.html>
7. N. Mitra (2003), *SOAP Version 1.2 Part 0: Primer*, W3C, URL – <http://www.w3.org/TR/soap12-part0/>, s1.1, p2
8. E. Christensen, F. Curbera, G. Meredith, S. Weerawarana (2001), *Web Services Description Language (WSDL) 1.1*, W3C, URL – <http://www.w3.org/TR/wsdl>, (section 1, page 2)
9. H. Thompson, D. Beech, M. Maloney, N. Mandelsohn (2001), *XML Schema Part 1: Structures*, W3C, URL – <http://www.w3.org/TR/xmlschema-1/>
10. P. Biron, A. Malhotra (2001), *XML Schema Part 2: Datatypes*, W3C, URL – <http://www.w3.org/TR/xmlschema-2/>
11. P. Biron, A. Malhotra (2001), *XML Schema Part 2: Datatypes*, W3C, URL – <http://www.w3.org/TR/xmlschema-2/#built-in-datatypes>, (section 3)
12. OASIS Web Service Security (WSS) Technical Committee, URL – <http://www.oasis-open.org/committees/wss>
13. A. Barbir, M. Gudgin, M. McIntosh (2004), *Basic Security Profile Version 1.0*, WS-I, URL – <http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html>
14. C Kaler, P. Hallam-Baker, R. Monzillo (March 2004), *Web Services Security: SOAP Message Security 1.0*, OASIS Standard 200401, URL – <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>
15. A. Nadalin, P. Griffin, C. Kaler, P. Hallam-Baker, R. Monzillo (March 2004), *Web Services Security: UsernameToken Profile 1.0*, OASIS Standard 200401, URL – <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf>
16. P. Hallam, Baker, C. Kaler, R. Monzillo, A. Nadlin (March 2004), *Web Services Security: X.509 Certificate Token Profile*, OASIS Standard 200401, URL – <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0.pdf>
17. D. Booth, C. Liu (December 2004), *Web Service Description Language (WSDL) Version 2.0 Part 0: Primer*, URL – <http://www.w3.org/TR/wsdl20-primer>, World Wide Web Consortium (W3C)

DISTRIBUTION LIST

Coalition Theatre Logistics (CTL)
Advanced Concept Technology Demonstration (ACTD)
Web Services Documentation

Kuba Kabacinski and Egon Kuster

AUSTRALIA

DEFENCE ORGANISATION

Task Sponsor

DGSL (CDRE Clint Thomas)	1
DDLIE (Mr Selby Dyer)	1

US CTL-ACTD Project Manager

Bruce A. Fink	3
---------------	---

S&T Program

Chief Defence Scientist	}	shared copy
FAS Science Policy		
AS Science Corporate Management		
Counsellor Defence Science, London		
Counsellor Defence Science, Washington		
Scientific Adviser to MRDC, Thailand		Doc Data Sheet
Scientific Adviser Joint		Doc Data Sheet
Navy Scientific Adviser		1
Scientific Adviser – Army		Doc Data Sheet & Dist List
Air Force Scientific Adviser		Doc Data Sheet & Dist List
Scientific Adviser to the DMO		Doc Data Sheet & Dist List

Information Sciences Laboratory

Chief Command & Control Division	Doc Data Sheet
Research Leader Command Decision	
Environments Branch	1
Research Leader Information Enterprises Branch	1
Research Leader Joint Command Analysis Branch	Doc Data Sheet
Research Leader Intelligence Information Branch	Doc Data Sheet
Head Human Systems Integration	Doc Data Sheet
Head Information Exploitation	Doc Data Sheet
Head Effects-Based Modelling and Analysis	Doc Data Sheet
Head Information Systems	1
Head Distributed Enterprises	Doc Data Sheet
Head Joint Operations Analysis and Support	Doc Data Sheet
Head Command Concepts and Architectures	Doc Data Sheet
Head Command Process Integration and Analysis	Doc Data Sheet
Head Intelligence Analysis	Doc Data Sheet
Egon Kuster (Task Manager)	1
Kuba Kabacinski (Author) c/o Egon Kuster	1
Publications and Publicity Officer, C2D/EOC2D	1 shared copy

DSTO Library and Archives

Library Edinburgh	1
Defence Archives	1
Library Canberra	Doc Data Sheet

Capability Development Group

Director General Maritime Development	Doc Data Sheet
Director General Capability and Plans	Doc Data Sheet
Assistant Secretary Investment Analysis	Doc Data Sheet
Director Capability Plans and Programming	Doc Data Sheet
Director Trials	Doc Data Sheet

Chief Information Officer Group

Deputy CIO	Doc Data Sheet
Director General Information Policy and Plans	Doc Data Sheet
AS Information Strategy and Futures	Doc Data Sheet
AS Information Architecture and Management	Doc Data Sheet
Director General Australian Defence Simulation Office	Doc Data Sheet
Director General Information Services	Doc Data Sheet

Strategy Group

Director General Military Strategy	Doc Data Sheet
Director General Preparedness	Doc Data Sheet
Assistant Secretary Governance and Counter-Proliferation	Doc Data Sheet

Navy

Maritime Operational Analysis Centre, Building 89/90 Garden Island Sydney NSW	Doc Data Sht & Dist List
Director General Navy Capability, Performance and Plans, Navy Headquarters	Doc Data Sheet
Director General Navy Strategic Policy and Futures, Navy Headquarters	Doc Data Sheet

Air Force

SO (Science) - Headquarters Air Combat Group, RAAF Base, Williamtown NSW 2314	Doc Data Sht & Exec Summary
--	--------------------------------

Army

SO (Science) - Land Headquarters (LHQ), Victoria Barracks NSW	Doc Data Sht & Exec Summary
SO (Science), Deployable Joint Force Headquarters (DJFHQ) (L), Enoggera QLD	Doc Data Sheet

Joint Operations Command

Director General Joint Operations	Doc Data Sheet
Chief of Staff Headquarters Joint Operations Command	Doc Data Sheet
Commandant ADF Warfare Centre	Doc Data Sheet
Director General Strategic Logistics	Doc Data Sheet

Intelligence and Security Group		
DGSTA Defence Intelligence Organisation		1
Manager, Information Centre, Defence Intelligence Organisation		1 (PDF)
Assistant Secretary Capability Provisioning		Doc Data Sheet
Assistant Secretary Capability and Systems		Doc Data Sheet
Assistant Secretary Corporate, Defence Imagery and Geospatial Organisation		Doc Data Sheet
Defence Materiel Organisation		
Deputy CEO		Doc Data Sheet
Head Aerospace Systems Division		Doc Data Sheet
Head Maritime Systems Division		Doc Data Sheet
Chief Joint Logistics Command		Doc Data Sheet
Head Materiel Finance		Doc Data Sheet
Defence Libraries		
Library Manager, DLS-Canberra		1
OTHER ORGANISATIONS		
National Library of Australia		1
NASA (Canberra)		1
UNIVERSITIES AND COLLEGES		
Australian Defence Force Academy		
Library		1
Head of Aerospace and Mechanical Engineering		1
Serials Section (M list), Deakin University Library		1
Hargrave Library, Monash University		Doc Data Sheet
Librarian, Flinders University		1
OUTSIDE AUSTRALIA		
INTERNATIONAL DEFENCE INFORMATION CENTRES		
US Defense Technical Information Center		1 PDF
UK Dstl Knowledge Services		2
Canada Defence Research Directorate R&D Knowledge & Information Management (DRDKIM)		1
NZ Defence Information Centre		1
ABSTRACTING AND INFORMATION ORGANISATIONS		
Library, Chemical Abstracts Reference Service		1
Engineering Societies Library, US		1
Materials Information, Cambridge Scientific Abstracts, US		1
Documents Librarian, The Center for Research Libraries, US		1
SPARES		3
Total number of copies:	34 printed copies +2 PDF	36

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA				1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)	
2. TITLE Coalition Theatre Logistics (CTL) Advanced Concept Technology Demonstrator (ACTD) Web Services Documentation			3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION) Document (U) Title (U) Abstract (U)		
4. AUTHOR(S) Kuba Kabacinski and Egon Kuster			5. CORPORATE AUTHOR Information Sciences Laboratory PO Box 1500 Edinburgh South Australia 5111 Australia		
6a. DSTO NUMBER DSTO-TN-0619		6b. AR NUMBER AR-013-356		6c. TYPE OF REPORT Technical Note	
7. DOCUMENT DATE March 2005					
8. FILE NUMBER 2004/1099766/1		9. TASK NUMBER 04/072		10. TASK SPONSOR DGSL	
				11. NO. OF PAGES 144	
				12. NO. OF REFERENCES 17	
13. DOWNGRADING/DELIMITING INSTRUCTIONS To be reviewed three years after date of publication				14. RELEASE AUTHORITY Chief, Command and Control Division	
15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT Approved for public release					
OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SA 5111					
16. DELIBERATE ANNOUNCEMENT No Limitations					
17. CITATION IN OTHER DOCUMENTS No					
18. DEFTEST DESCRIPTORS Joint Logistics Logistics Information Systems Computer Architecture					
19. ABSTRACT This document describes the development and utilisation of web services in the Coalition Theatre Logistics Advanced Concept Technology Demonstrator (CTL ACTD) prototype. Included is a background on web services, issues with using web services, and technical details and specifications of all services used in CTL ACTD. The appendices contain sample Web Service Description Language (WSDL) files and the current XML Schema definition. Information contained within captures current implementation details and can be used as a reference for future implementation. This document does not convey details of the operational CTL Spiral 1 and 2 currently deployed as part of JP2030's Movement Management System (MMS). Only details about the CTL ACTD prototype system are covered.					