10^{th} INTERNATIONAL COMMAND AND CONTROL RESEARCH AND TECHNOLOGY SYMPOSIUM THE FUTURE OF C2

Using XML-based Web Services to Implement a Prototype C2 System

Topic：Experimentation

Authors：Ching-Show Lin, Chia-Hao Liang (*)

POC：Chia-Hao Liang

Computer Science Department, Chung-Cheng Institute of Technology, National Defense University

No. 190, Sanyuan 1st St., Dashi Jen, Taoyuan, Taiwan 335. R.O.C.

Tel：886-3-3805249#310

Fax：886-2-3894770

Email:g941202@ccit.edu.tw or cslin@ccit.edu.tw

Abstract

Command and Control system is a huge and complex system of systems. For its ability to improve the command and control efficiency and multiply operational capability, C2 system investment is always a benchmark for military modernization. However, most of C2 systems were independently developed, validated and approved as a stand-alone solution to reflect service requirement rather than joint focused. These stovepipe systems not only have an adverse impact on joint or coalition operation but also are fairly difficult to integrate and interoperate effectively with other systems. To promote the integration and interoperability between these stovepipe legacy systems, the paper applies XML standard to redefine the structured radar track and GPS positioning data formats whose original messaging formats are assumedly USMTF and GPRMC. And, we also implement radar tracking data and GPS positioning data generator to simulate air and land targets. In addition, a static intelligence databases such as order of battle are built for information exchange with other systems. Sensors, GPS and intelligence Web Services including SOAP and WSDL are constructed to provide near real-time static intelligence and dynamic track information services. All relevant C2 centers may subscribe the necessary services from the providers to work together with their own systems for mission needs. The implementation result demonstrates XML-based

Web Services technology and makes C2 system integration easy, flexible and cost effective.

1. Introduction

To promote operation capability, many countries have invested a great deal of time and resources to establish command and control (C2) systems. Using information strength achieves decision-making superiority. C2 system integrating operation, intelligence, logistics and personnel functions across the different organizations and agencies is a huge and complex system of systems. Over the past years, C2 systems were independently developed, validated and approved as a stand-alone solution to reflect service requirement rather than joint focused. To be effective, organizations must share information across dissimilar, independent developed systems that involve diverse languages, cultures and command structures. Consequently, the integration of heterogeneous deployed C2 systems, called stovepipe systems, should be prioritized to implement for the purpose of enhancing inter-service interoperability and information sharing between different systems.

The typical representatives of system integration technology are OMG CORBA, Microsoft DCOM and Sun RMI. CORBA specifies a system which provides interoperability between objects in a heterogeneous, distributed environment and in a way transparent to the programmer. CORBA adopts Object Request Broker (ORB) and Internet Inter-ORB Protocol (IIOP) to communicate with the other objects. The clients request services from objects through a well-defined interface. This interface is specified in Interface Definition Language (IDL). A client accesses an object by issuing a request to the object. The request is an event, and it carries information including an operation, the object reference of the service provider, and actual parameters (if any). The object reference is an object name that defines an object reliably [1]. Figure 1 shows the concept of CORBA.

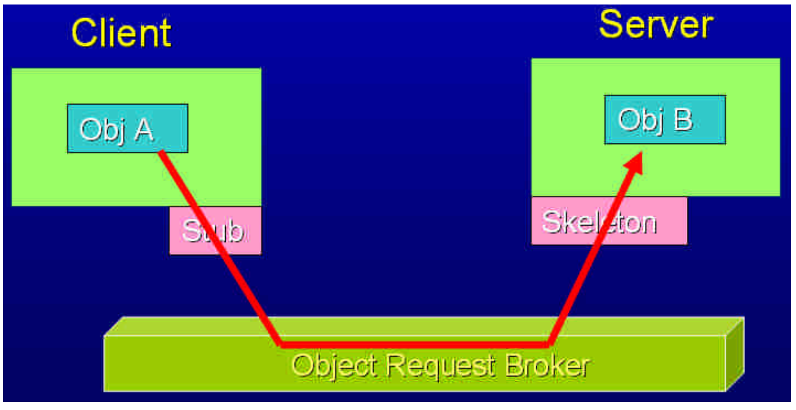

Fig 1. Concept of CORBA

However, The traditional distributed computing technologies have the following disadvantages: (1) CORBA or DCOM are too complex to implement. (2) Unless the objects are in the same architecture, otherwise interoperability problems can not be avoided. (3) These technologies are only applied for Intranet rather than Internet which has a firewall mechanism to deter the entrance of machine codes CORBA or DCOM generates. As a result, the application of CORBA or DCOM had the limitation [2].

For joint or coalition operation needs, any particular service system should interoperate with other service systems, agency systems and allies systems for information sharing. XML technology provides a standard for data interchange. XML is a data markup language like HTML, using bracketed tags to describe the structured data. With HTML, the tags primarily relate to the formatting and display of the text. With XML, the tag is extensible, anyone can define a tag to describe the attribute of the text. If applications agree on their use of the tag definitions, then they can understand the context of the text exchanged between them. Consequently, the use of XML is particularly well suited for web environments and allows the information to be both human and machine readable and is able to enhance the integration of system in a distributed computing environment. Based on XML technology, web services are software components that are loosely coupled, encapsulated and that can be contracted using standard Internet protocols. Web services are able to facilitate web-based system integration in an easy and cost effective way.

In recent years, MITRE Corporation, a US government-owned non-for-profit and development corporation, has made much more efforts on information exchange between C2 systems using XML technology. The legacy message text formats (MTF) have been transferred into XML-MTF in order to implement automated decision making processes of military operations [3]. MITRE also helped US military to establish a Joint Battlespace Infosphere (JBI), called information supermarket, where information providers submitted the data to JBI, and information consumers extracted data they required from the JBI. The goal of JBI is to provide the right information at the right time so that commander can do the right things at the right time in the right way [4]. The core technologies of JBI are XML and web services.

In the C2 integration and interoperability issue, ROC armed force also faces the same challenges as US. In this paper, we try to adopt these new technologies to implement a prototype C2 system. The system will integrate air track, land track and static intelligence database services to generate a common operating picture (COP) in a network environment.

The remainder of this paper is organized as follows：In Section 2, we give an overview of our prototype C2 system architecture and an introduction of web service architecture. In Section 3, we articulate the steps of system implementation and lessons learned. Finally, the conclusion is drawn in Section 4.

2. System Overview

2.1 Prototype system architecture

Our prototype system consists of several main components：track generator, track web service, intelligence database and client view. The system architecture is shown in Figure 2. These components are described as follows：

Track generator：For simulating the physical radar track and GPS track, we use a track generator to produce radar and GPS tracks periodically whose formats are assumedly USMTF and GPRMC. The purpose is to demonstrate the application of data exchange by transferring USMTFs and GPRMC formats into XML formats.

Track web service：We implement a track web service to provide consumers such as C2 centers to access the track data from tracks database via SOAP and WSDL technologies. The track web service can be dynamically interfaced and invoked by other component for their mission needs.

Intelligence web service：The intelligence web service is to provide consumers to query the static intelligence data such as order of battle and topography from the common intelligence database.

Client view：The client view is to integrate the track and intelligence web services distributed in a network. The track and intelligence information is shown in Graphic User Interface with digital map. The client or C2 center can integrate the different web services and redesign them for their mission requirements.

In our implementation, Microsoft IIS 5.0 is used as Web Server and database system is Microsoft SQL Server 7.0 desktop. Case tool for web services and client application development is Microsoft Visual Studio 2003 and programming language is Visual Basic .NET.

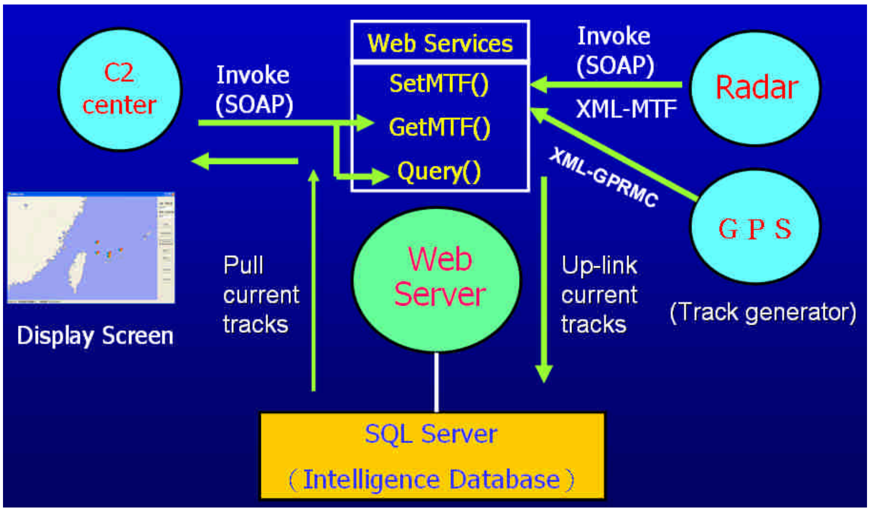

Fig 2. System Architecture

2.2 Web Services Architecture

W3C describes that web service is a software component designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using popular HTTP protocol with an XML serialization in conjunction with other Web-related standards [5].

There are four main components in web service operation as shown in Fig.2:

1. **eXtensible Markup Language (XML)**: XML is the base of Web Services. It is a markup language adding a tag to the context of data. Unlike HTML, XML is a flexible, self-defined and well-structured language. When XML is parsed by a parser, data can be extracted easily.
2. **Simple Object Access Protocol (SOAP)**: SOAP is a XML protocol for web services. SOAP provides a service-oriented architecture for server-to-server and server-to-device communication via a variety of underlying protocols [6]. SOAP likes an envelope which contains the messages what we want to send, and also provides binding information of invoking web service.
3. **Web Services Definition Language (WSDL)**: WSDL is similar to IDL of CORBA and provides the interface of Web Service. WSDL describes the name, function, parameters, location and data type of services in a XML format.
4. **Universal Description, Discovery Integration (UDDI)**: UDDI is a registry mechanism and used for the advertisement, cataloging and integration of web services.

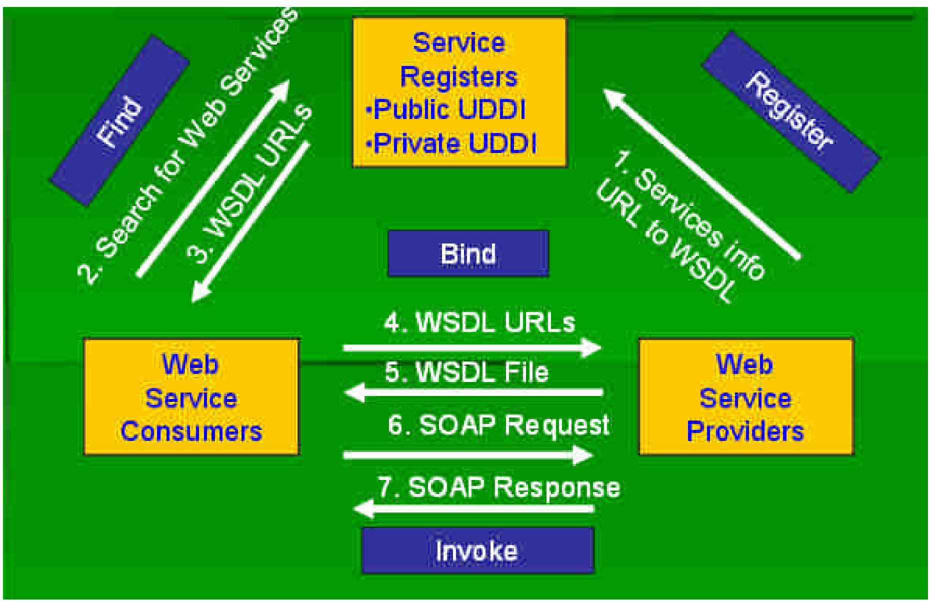

Fig 2. Illustration of Web Service Operation

2.3 .NET Framework

Microsoft mentions that .NET is software connecting information, people, systems and devices. It spans clients, servers and developer tools, and consists of: .NET framework used for binding and running all kinds of software, IDE tools like Visual Studio, Windows Server, SQL Server and client side likes Windows XP [7]. .Net framework supports over 20 different programming languages and makes it easier to build, deploy and has more secure, robust and high-performing applications than before. .NET framework is composed of the common language runtime (CLR) and a unified set of class libraries. CLR is the key component of the .NET framework. CRL has the same concept as JAVA virtual machine, programs with common intermediate language (CIL) including VB and C# and compile into Microsoft Intermediate Language (MSIL). This is the reason why .NET can solve interoperability problems of system components developed by different programming languages.

2.4 ADO .NET

Database access is an important part to implement the prototype system. Real time air and land tracks are stored temporarily in the intelligence database, and C2 centers pull radar tracks from the database through ADO .NET via response SOAP message. ADO, stands for Active Database Object, is a Microsoft technology and has three advantages: (1) all data in ADO .NET is transported in XML format, so it can be read in any platform. (2) ADO .NET promotes use of disconnected datasets, with automatic connection pooling bundled as part of the package. (3) ADO .NET adopts disconnected dataset, the database server is no longer a bottleneck and hence applications should incur a performance boost [8]. ADO .NET consists of dataset and data provider. And, data provider contains Connection, Command, Data Reader and Data Adaptor. Data provider is responsible for connecting database, and pushes required data to dataset. Dataset is a data container containing a collection of one or more Data Table objects made up of rows and columns of data. DataSet is also a memory-resident representation of data and can provide a consistent relational programming model regardless of the source of the data. Fig 3 shows the concept of ADO .NET Architecture [9].

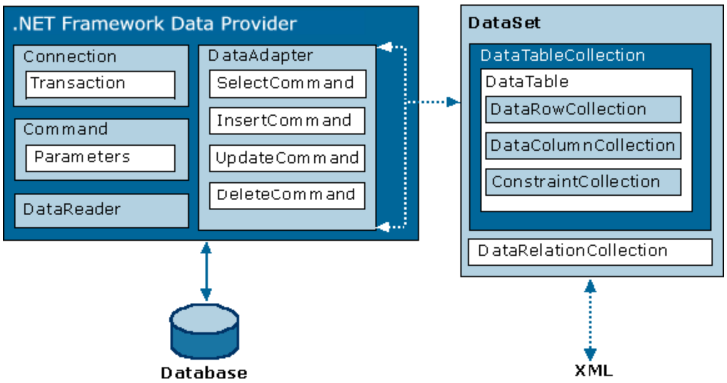

Fig 3 ADO .NET Architecture

3. System Implementation

3.1 Assumption

For the purpose of implementation, we make the following assumptions:

(1) The radar can provide the target tracks periodically.

(2) GPS system can receive the positioning data periodically.

(3) The radar track formats are US MTF and stored in a text file.

(4) GPS positioning formats are GPRMC and stored in a text file.

(5) The agreement on the tag or element names in XML formats has been reached.

3.2 XML-USMTF Transfer

The US Message Text Format (USMTF) is the standard format having been used by the joint forces and coalition allies for 20 years. They support the full spectrum of military operations, including air, intelligence, logistic and medical operations. These legacy messaging standards have proprietary formatting rules and make them difficult to leverage commercial off-the shelf (COTS) tools for message processing and component integration support. The birth of XML technology has taken the world by storm, only adding the tags in MTF using XML standard can improve the ability to find, retrieve, process and exchange large amounts of information across systems, organizations and formats boundaries. XML technology makes system integration and interoperability easier.

Recently, MITRE has made much more efforts in XML-MTF schema and document specifications for use by the US military and joint forces. Here, we reuse this method to transfer MTF into XML format for data exchange. Our prototype system integrates air tracks in our C2 systems. Assume the original air track format is MTF, we take the following example to analyze:

AIROP/020200Z/6/US/FTR/F15/TN:401/LM:2300N12300E/CRS:180/SPD:600KPH/ALT:12000FT//

After adding the tag to name each data element of MTF, the result is shown as follows:

```xml
<?xml version="1.0"?>
<Operation>
  <op_type type="Air Operation">
    <DateTime>020200Z</DateTime>
    <Quantity>6</Quantity>
    <Country>US</Country>
    <Catalog>FTR</Catalog>
    <Type>F15 </Type>
    <TrackNumber>401</TrackNumber>
    <Axis>2300N12300E </Axis>
    <CRS>180</CRS>
    <Speed>600</Speed>
    <Altitude>12000</Altitude>
  </op_type>
</Operation>
```

When we add the tags in MTF message, C2 system can understand the number ‘6’ means the quantity of aircraft via XML parser.

3.3 XML-GPS Transfer

Besides air track, we would like to track the positioning data of mobile vehicles such as tanks from GPS. Therefore, we need to transfer GPS message into XML format. Assume the original positioning format is GPRMC, which stands for Recommended Minimum Specific GPS/TRANSIT Data**.** The GPRMC format is shown in table 1. [10].

Table 1. GPRMC format

$GPRMC,<1>,<2>,<3>,<4>,<5>,<6>,<7>,<8>,<9>,<10>,<11>,<12>*hh	
1	UTC of position fix. (hhmmss.sss)
2	Data status (V=navigation receiver warning, A= locate successfully)
3	Latitude of fix. (ddmm.mm)
4	N or S
5	Longitude of fix.(dddmm.mm)
6	E or W
7	Speed over ground in knots
8	Track made good in degrees True. (0.0~359.9)
9	UT date (ddmmyy)
10	Magnetic variation degrees. (000.0~180.0)
11	E or W

12	Checksum

The following is an example of GPRMC message:

$GPRMC,161229.487,A,25.0377,N,121.3366,E,0.13,309.62,120598,1.2,E ,*10

Similarly, we also transfer GPRMC into XML called XML-GPRMC. The result is shown as follows：


```
<?xml version="1.0"?>
<Operation>
  <op_type type="GPRMC">
    <UTC>161229.487</UTC>
    <Status>A</Status>
    <latitude>25.0377</latitude>
    <dir_of_lat>N</dir_of_lat>
    <longitude>121.3366</longitude>
    <dir_of_long>E</dir_of_long>
    <TrackDegree>0.13</TrackDegree>
    <UT_Date>120598</UT_Date>
    <MVG>1.2</MVG>
    <dir_of_MVG>E</dir_of_MVG>
    <Checksum></Checksum>
  </op_type>
</Operation>
```


3.4 Web Server Side

The server side of the system is responsible for providing all kind of services such as radar tracks, land tracks and intelligence data. Web server is a service provider and can be distributed at everywhere in a network. Only the consumers find the location of web service via UDDI, they can request the service via WSDL and SOAP message.

3.5 Client Side

The clients or users are the web service consumers such as C2 centers. The C2 centers can request the web services they need distributed in a network. In this study, C2 centers invoke the air tracks, land tracks and intelligence web services to be displayed in GUI. The C2 centers can have the common operating pictures and intelligence supports and are able to command and control. In addition, the different level of C2 centers can integrate the web service with their own legacy systems according to their mission requirements. .

4. Result Analysis and Lesson Learned

4.1 Result Analysis

In this section, we use a scenario to demonstrate system functions in client side as shown in Fig 4. First, we produce thirteen air tracks (including seven red Hs and seven blue Fs symbols) and one land track (green dot symbol) randomly and update the track data every 5 seconds. Second, we send them to database for storage and keep the track data new. Then, we start to retrieve track web service from database by clicking the “Retrieve” button, the responded track data flows into client application program. Third, client application program begins to interpret the track data and displays the symbol, identification number, speed and heading on digital map. Finally, we query the radar coverage service from intelligence database and show blue circles on the screen. The client side also can choose on or off switch to decide whether to trigger the track or intelligence service according to their own requirements.

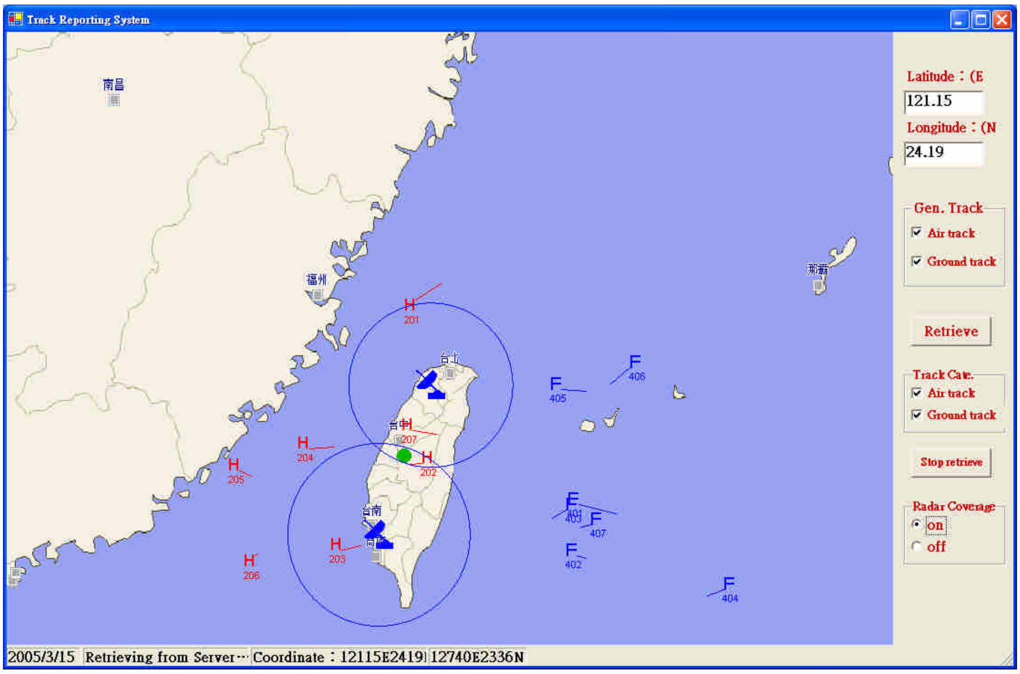

Fig 4. Client View of System

4.2 Lesson learned

From the implementation result, we can see that XML is useful and easy to implement for tactical data exchange. In particular, the legacy system can also become web service producers via XML representation. The basics of XML are fairly simple. This is one of XML’s great strengths. The most difficult part is getting the developers of disparate systems to agree on the tag or element names. This is often a painful process and independent of technology. Often it will spend a lot of time to reach this agreement among the relevant organizations. For the purpose of demonstration, the paper just uses radar tracks and GPS positioning messages, in fact the other information related to warfighting also can follow the same method to become web service providers. The consumers can make their own value-added applications by invoking web services distributed in a network.

5. Conclusion

Before XML and web services, integration was highly difficult and expensive occurring either at data, application and process level. The paper demonstrates XML can promote the data exchange capability no matter what its original format is. Only adding the tag of data element, the data becomes easy to be recognized. This paper also shows XML based SOAP and WSDL web service technologies not only facilitate the integration of new and legacy systems but also enhance system interoperability. In our prototype system implementation process, C2 centers can dynamically integrate the various sensor and intelligence web service to meet their mission needs no matter where these web services are distributed in a network. Accordingly, the commander of C2 center has the situation awareness capability and can make the right decision at the right time in the right way. In the foreseeable future, the use of XML based web services in system development and integration aspects will become the mainstream, not only can save money but also can reduce training time.

.

References

[1] *A Brief Tutorial on CORBA*, http://www.cs.indiana.edu/~kksiazek/tuto.html.

[2] Schmelzer, Vandersypen, Bloomberg, “XML and Web Services, ” SAMS, Feb 2002.

[3] Robert D. Cherinka, Robert W. Miller, “*Using XML-based Web Services to Enhance Information Interoperability in the Department of Defense*,” 2001

[4] US Air Force Scientific Advisory Board, *Building the Joint Battlespace Infosphere*, Dec. 1999.

[5] W3C Working Group, *Web Service Architecture*, 2004.

[6] W3C Working Group, *SOAP 1.2 Part 1: Messaging Framework*, 2003.

[7] *Technology Overview*, http://msdn.microsoft.com/netframework/technologyinfo/overview/default.aspx.

[8] *ADO Overview*, http://www.developer.com/net/vb/article.php/10926_1540311_2

[9] *ADO .NET Architecture*, http://msdn.microsoft.com/library/default.asp.

[10] NMEA 0183, http://www.kh-gps.de/nmea-faq.htm.