Leveraging Service Oriented Architecture:
A Case Study for Ocean Energy Information Management

Ante Bosnjak , Shihong Huang, James J. Mulcahy
Dept. of Computer & Electrical Engineering and Computer Science
Florida Atlantic University
(abosnjak,shihong,jmulcah1)@fau.edu

Abstract

Often Service Oriented Architecture (SOA) is implemented after an organization has an already established legacy set of applications and data formats. SOA is a powerful approach in that it can encapsulate and streamline data into a format that can be understood by an entire organization. From the executive department to the field office, data can be universally accessible. If there is a need to gather, consume, and create reports from data stored in a MySQL database at the field office with an Oracle database from the accounting department and a Progress database at the manufacturing plant, then you need SOA. This interconnection is not exclusive to databases, but rather any form or type of data for which there is a need to gather and compare data. Whether it is stored in various file formats from a Linux server, Solaris server or Windows server to an embedded device, then you need a SOA. This paper is an analysis of one SOA approach and the basic underlying network layers of it.

Choosing and subsequently implementing a service oriented architecture for a large organization can be a complex undertaking, but Swordfish and Eclipse SOA, the new emerging SOA framework and integrated development environment, could mediate this. In this paper the eclipse SOA and Swordfish runtime framework are investigated and presented as a viable open source service oriented architecture solution for the enterprise.

Keywords: Service Oriented Architecture (SOA), Web Services, Enterprise Applications, Distributed Systems, Eguinox, OSGI, Swordfish, Eclipse, Eclipse SOA, Apache ServiceMix, Apache CFX.

1. Introduction

Large organizations are tediously involved with the creation, compilation and use of data by internal departments, constituents and external entities. When data or information needs to be collected and compiled by members of such an organization, the relevant data or information may traverse data acquired by various members of that organization.

In such an organization it is inevitable for a diversity of information and data formats to exist. Consequently various unrelated software systems are used by different members to maintain, store and collect their information and data. Because of this, data needed to be translated into useful information by various members of different departments or entities within the organization. Information needs to flow through an organization as fast as a person could send, receive, translate and interpret it.

The solution to information bottlenecks is to create a networked enterprise within the organization that facilitates the accessibility, mutability and persistence of data. Such a networked architecture on computers within and across an organization would allow information to flow instantaneously and uninhibited without a need for members to translate data from one format into another.

The project described in this paper aimed to facilitate the integration of the diverse collection of data of an organization, and combining unrelated software systems into a decentralized architecture that can manage and facilitate the sharing and updating of information among its vast user base.

The goal is to accommodate the various departments of an organization with a dynamic, decentralized and functional service oriented architecture that can be utilized by members of the organization or even composite web services. This service oriented architecture will allow members of an organization to access, store, and compile information from various data formats in a fluid manner across organizational boundaries.

IEEE IRI 2011, August 3-5, 2011, Las Vegas, Nevada, USA
978-1-4577-0966-1/11/$26.00 ©2011 IEEE

Section 2 describes the core concepts and technologies that are involved in a service oriented architecture. Section 3 gives a general overview of the organization's layout and its requirements for the implementation of a SOA. Section 4 gives an in-depth introduction into the service oriented architecture behind Eclipse SOA and Swordfish. Section 5 describes some initial experiments in developing and running web services with Eclipse SOA, as well as subsequent deployment into a deployable Swordfish server framework. Section 6 concludes information from the effort gathered thus far, summarizing the workability, feasibility and possible realization of the Eclipse service oriented architecture system for the Southeast National Marine Renewable Energy Center (SNMREC), a U.S. Department of Energy designated national center for research at Florida Atlantic University.

2. A Decentralized Service Oriented Architecture

For years the internet has been characterized by web traffic and portals, but this paradigm is now changing into one of automated and complex electronic transactions. The internet or local networked communication has become a more integral part of the application domain. Eventually there has arisen a need for applications to invoke programs remotely across a network.

At first, programs were created on client machines that executed remote procedure calls to applications running on servers across a network or on the internet. The problem with this approach is that client and server applications were either centralized or hardwired to one another. If the server application changed then all clients needed to be changed or updated in order for the remote procedure call over the network to function properly.

Over time this led to the need for an open specification which could standardize an application communication layer. Thus XML-RPC and SOAP specifications were formalized. Services provided by an application running on the server would be accessible by any client, running on any platform. Data and communication exchange over the internet is made possible by a protocol suite of communication layers.

The fundamental component for an organization's networked enterprise is the application layer. Below this layer is the network layer (IP), and the transport layer (TCP or UDT). The application layer resides above these two layers and is an assemblage of subsequent layers on the communication protocol stack. Collectively this combination of layers - network, transport and application – is commonly known as the internet layer. What is most relevant is the application layers an important and fundamental part of an SOA.

Much of the logic of an SOA resides in this application layer. Those layers which are added to the communication protocol stack's application layer create a service oriented architecture (SOA). These communication layers are responsible for the transport and exchange of data across the internet and networks in XML-compliant structures. An SOA is a fairly complex mix of open standards specification protocols such as SOAP, UDDI, WSDL, and numerous WS* specifications (those are second generation web service specifications – WS-BPEL ...etc.) [3][6][16].

These open standards specifications make asynchronous communication possible between autonomous services running on an organization's network enterprise. Loosely coupled service architecture is what differentiates SOA from other network architectures.

Communications within an SOA is message based. Messaging can be described as the transfer of data stored in XML documents through the SOAP (Simple Object Access Protocol) protocol specification. SOAP provides for the serialization, transport and de-serialization of XML documents. XML schema allows data contained in messages to be represented in an established and standardized format that is defined by an XML data schema.

This XML schema subsequently requires and defines an XML document which attaches meaning and context to its stored data, thereby preserving the integrity and validity of the data. SOAP uses many protocols once such protocol is the HTTP protocol which it uses to transport these XML documents from one system to another regardless of the operating system, application programming interface libraries or object model framework used. As long as systems understand the SOAP protocol, they can send and receive data in a way that is independent of the platform.

The importance of the SOAP protocol is that it enables data encapsulated within XML documents to exchanged between web-based distributed services (web services). However, SOAP's standardized communications framework is merely a means of information transport, and does not have a means to convey the functionality of the web services. The XML-

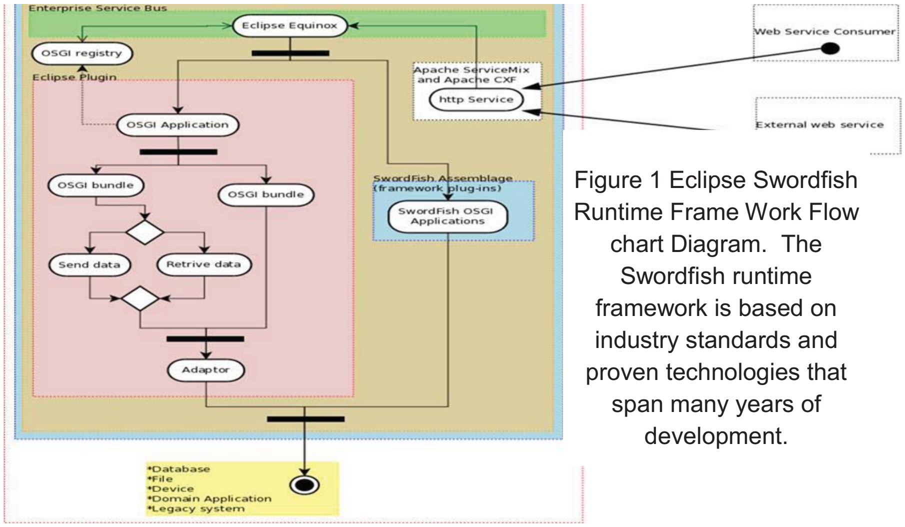

Figure 1 Eclipse Swordfish Runtime Frame Work Flow chart Diagram. The Swordfish runtime framework is based on industry standards and proven technologies that span many years of development.

based language WSDL (Web Service Description Language) addresses this issue. WSDL describes SOAP-enabled services. It is used to describe the details and mechanics of invoking a web service and its public interfaces in a platform-independent and language-independent way. Web services allow organizations to bridge the gaps between heterogeneous and disparate application environments. SOA is entirely based on web services and other complementing specifications such as UDDI that facilitate the registry and discovery of web services. Figure 2 is a graphical representation of a service oriented architecture.

3. The Southeast National Marine Renewable Energy Center Legacy Systems

The Center of Ocean Energy Technology at FAU is a large organization involved with work that spans departments, constituents and external entities. The data and research is utilized by a large user base comprised of teachers, students, researchers, employees, government officials, and industry clients.

The center needed a service oriented architecture that could leverage data availability and accessibility. The existing system currently is comprised of researchers depositing raw data on either the local, departmental or center’s computer infrastructure. There is currently no networked organizational enterprise system that allows members of the center to access, store, and compile information from various data formats in a fluid manner across the organizational boundaries of the center.

4. Eclipse SOA and Swordfish Architecture

Implementing an SOA system for the SNMREC project is an endeavor that may ultimately span years of research and development. The SOA system for the SNMREC project is expected to be dynamic and robust enough that it can accommodate an ever-growing and changing domain.

An optimal solution is an extensible SOA framework based upon proven, established and emerging open standards. The Eclipse Swordfish runtime and SOA platform, although still in the incubation stages of development at Eclipse.org, is the best choice for an open standards-based SOA framework [13][14] .

The Swordfish SOA Runtime Framework (Figure 2) is a tested, tried and proven SOA framework. Swordfish is derived from Germany's Dueche Post (postal service) SOA framework that has been open sourced and contributed to the Eclipse foundation through Sopera [15]. The advantage of using Swordfish is open source and part of the Eclipse foundation's SOA platform. The benefit of the Eclipse Foundation is that it is an open source software platform that is backed and supported by major software industry leaders such as IBM, Red Hat, and Borland, among others, ensuring that this software platform will be around for years with active development, updates and bug fixes.

Swordfish is the Eclipse SOA's runtime framework. It has been integrated into the Eclipse SOA Platform which is a runtime and tools integration platform for SOA developers. The Swordfish runtime framework that is primarily based on Eclipse Equinox and OSGI (Open Services Gateway Initiative) [5] is used by NASA's Maestro and Ensemble [9] projects. It is used as a space mission software platform in an Eclipse RCP (Rich Client Platform) [4] deployment. OSGI is a powerful modular framework and runtime whose specifications are produced by the OSGI Alliance, an industry consortium of software and embedded systems vendors such as IBM, Red-Hat, Hitachi, Oracle, NEC, Siemens and Ericsson. Eclipse Equinox is a robust scalable and comprehensive implementation of OSGI [5][11] specifications. Other technologies the Eclipse Swordfish framework incorporates is the inclusion of proven open source components such as Apache ServiceMix [10] and Apache CXF [12]. Apache CFX is an open source web services framework that has binary and legacy protocol support. Apache ServiceMix is an enterprise service bus with functionality that is a combination of event-driven architecture and service oriented architecture.

Some of the technologies that the Eclipse Swordfish SOA framework is derived from are used to manage the

Spirit and Opportunity missions on Mars. If these underlying technologies that make up the Eclipse Swordfish SOA framework are viable and robust enough for NASA to conduct missions on Mars, then service oriented architecture derived from and based upon such technologies should likewise be a viable and robust solution for the SNMREC project.

Another benefit of the Swordfish runtime framework is that it is included in the Eclipse SOA Platform. This is a runtime and tools integration platform for SOA developers. It simplifies the development of SOA and the testing of services by being able to execute them on an all-inclusive platform. The Eclipse SOA Platform includes a Java IDE, Swordfish Tooling, ECF's OSGi 4.2 remote services implementation, the Plugin Development Environment (PDE), an XML Editor, and a WSDL Editor.

5. Experimental Results

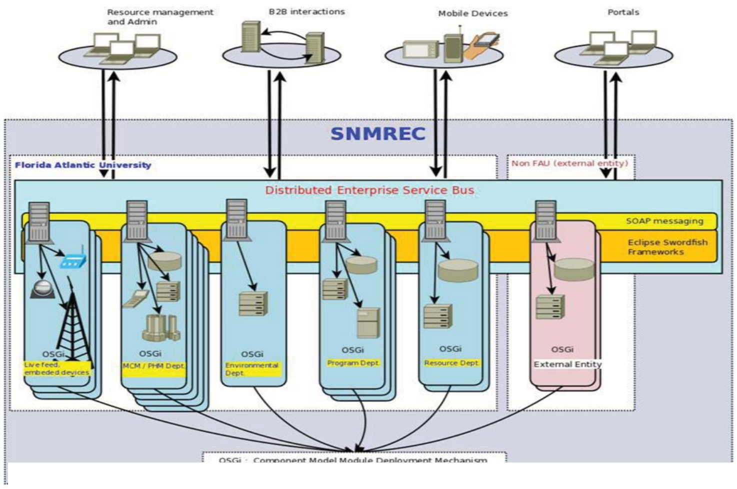
Resource management and Admin
B2B interactions
Mobile Devices
Portals
SNMREC
Florida Atlantic University
Non FAU (external entity)
Distributed Enterprise Service Bus
SOAP messaging
Eclipse Swordfish Frameworks
OSGi
Live feed, embeded devices
OSGi
MCM / PHM Dept
OSGi
Environmental Dept
OSGi
Program Dept
OSGi
Resource Dept
OSGi
External Entity
OSGi : Component Model Module Deployment Mechanism

Figure 2 Swordfish OSGi framework, a dynamically updateable SOA for a large organization such as SNMREC

The implementation of SNMREC's SOA is being developed using the SOA Eclipse IDE and tooling. Implementation of web services requires creating either a BPEL project in Eclipse or manually creating a WSDL file for a Web Service. The next step involves creating a JAX-WS Service from a WSDL file. This service can either be a consumer or a provider. Composite services can then be created from a combination of these services. These services are created as Eclipse OSGI plugins [2]. Web services are then run and tested internally on the Eclipse SOA development system. It has an integrated Swordfish runtime which includes all the Swordfish OSGI bundles and dependencies. After native testing, the web service is exported as a plugin. This plugin was then deployed as an OSGI application bundle into a standalone Swordfish server runtime. This process comprised the testing and development phases of a Web Service for the SOA.

The implementation of a UDDI is currently still under development. For the purpose of testing the Eclipse SOA tooling, a prototype example of a reservation system was created as a composite web service. It was composed of two separate web services - a payment verification service and reservation number verification system.

The experiments and results gathered thus far are limited to a preliminary investigation for a viable SOA solution for use at SNMREC. No actual or functional domain-specific web services have been created. However, the results prove the ease, simplicity and quick implementation time of developing web services for the SOA with the Eclipse SOA IDE and the Swordfish framework. Experiments with the deployment and development of specific web services for the SNMREC SOA needs to be addressed with further testing of actual data and applications which are specific to SNMREC's domain.

6. Conclusions

An analysis of the Eclipse SOA and Swordfish runtime framework was necessary for the newly formed national organization (SNMREC) so that it could understand the logistical possibilities offered by service oriented architecture.

We wanted to research and test a potential SOA for the SNMREC organization. There are many different SOA integrations available for SNMREC organization to choose. We chose to experiment and conduct an analysis of the Eclipse SOA for SNMREC primarily because it is an open source solution and is standards-based.

We decided that the Eclipse SOA tooling and its Swordfish runtime framework for deployable web-services would be a viable solution for the SNMREC organizational SOA. However, features such as a central UDDI repository, security and access restriction OSGi plugins for the Swordfish framework will need to be developed and implemented in order to make this a completely viable SOA for the organization. Because this system is open source, modifying and developing in-house solutions to the unique requirements of the organization can be done. Furthermore, the industry-backed open standards components of the Eclipse SOA and the Swordfish runtime ensure system interoperability and future development.

7. Future Work

Future work on the SNMREC project involves developing both java OSGI plugins for the Swordfish server that add new functionality and department-specific web services. Developing a UDDI plugin for the Swordfish runtime framework will be the first priority.

Acknowledgments

Work on this project is funded by Southeast National Marine Renewable Energy Center (SNMREC), a U.S. Department of Energy designated national center for research at Florida Atlantic University.

References

[1] Brogden, William B. “SOAP Programming with Java”. San Francisco: Sybex, 2002. Print.

[2] Clayberg, Eric, and Dan Rubel. “Eclipse Plug-ins (3rd Edition)”. Addison Wesley. Print.

[3] Erl, Thomas. “Service-Oriented Architecture (SOA): Concepts, Technology, and Design”. Prentice Hall/PearsonPTR, 2010. Print.

[4] McAffer, Jeff, Jean-Michel Lemieux, and Chris Aniszczyk. “Eclipse Rich Client Platform”. Upper Saddle River, NJ: Addison-Wesley, 2010. Print.

[5] McAffer, Jeff, Paul VanderLei, and Simon Archer. “OSGi and Equinox: Creating Highly Modular Java Systems”. Upper Saddle River, NJ: Addison-Wesley, 2010. Print.

[6] Papazoglou, Michael P. “Web Services: Principles and Technology”. Harlow: Pearson Prentice Hall., 2008. Print.

[7] Reilly, David, and Michael Reilly. “Java Network Programming and Distributed Computing”. Boston, MA: Addison-Wesley, 2002. Print.

[8] Linthicum, David S. “Next Generation Application Integration: from Simple Information to Web Services”. Boston: Addison-Wesley, 2004. Print.

[9] “NASA Uses Eclipse for Interplanetary Operations” http://www.eclipse.org/community/casestudies/NASAfinal.pdf

[10] "Apache ServiceMix, the Agile Open Source ESB -- Index." Apache ServiceMix 4.0. Apache Software Foundation. Web. 2010.<http://servicemix.apache.org/SMX4/index.html>.

[11] "Turning Ordinary OSGi Services Into Remote OSGi Services | OSGi Zone." OSGi Zone | Community Resources for OSGi Developers and Users. Web. 2010.<http://osgi.dzone.com/articles/turning-ordinary-osgi-services>.

[12] "Java Web Services: Introducing CXF." IBM – United States. Web. 2010. <http://www.ibm.com/developerworks/java/library/j-jws12.html>.

[13] "Learn about Swordfish." Eclipse - The Eclipse Foundation Open Source Community Website. Web. 2010. <http://www.eclipse.org/swordfish/learn/learn.php>.

[14] “Eclipse Swordfish — an Open Source SOA Runtime. Framework for the Enterprise.” Whitepaper. Oliver Wolf (SOPERA GmbH), February 2009 <http://www.eclipse.org/swordfish/assets/whitepaper.pdf

[15] "Eclipse Reeling in Swordfish as SOA Framework | Platforms - InfoWorld." Business Technology, IT News, Product Reviews and Enterprise IT Strategies - InfoWorld. Web. 14 Dec. 2010. <http://www.infoworld.com/t/platforms/eclipse-reeling-in-swordfish-soa-framework-228>

[16] Valipour, M.H.; Amirzafari, B.; Maleki, K.N.; Daneshpour, N.; , "A brief survey of software architecture concepts and service oriented architecture," Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on , vol., no., pp.34-38, 8-11 Aug. 2009 doi: 10.1109/ICCSIT.2009.5235004 URL: http://ieeexplore.ieee.org.ezproxy.fau.edu/stamp/stamp.jsp?tp=&arnumber=5235004&isnumber=5234374

[17] Lewis, G.A.; Morris, E.; Simanta, S.; Wrage,L.; , "Common Misconceptions about Service-Oriented Architecture," Commercial-off-the-Shelf (COTS)-Based Software Systems, 2007. ICCBSS '07. Sixth International IEEE Conference on , vol., no., pp.123-130, Feb. 26 2007-March 2 2007doi: 10.1109/ICCBSS.2007.9 URL: http://ieeexplore.ieee.org.ezproxy.fau.edu/stamp/stamp.jsp?tp=&arnumber=4127307&isnumber=4127277

[18] Qin Li; Huibiao Zhu; Jifeng He; , "A Formal Perspective for Service Coordination Framework in Service Oriented Architecture," Software Engineering Conference, 2009. ASWEC '09. Australian , vol., no., pp.287-296, 14-17 April 2009 doi: 10.1109/ASWEC.2009.11 URL: http://ieeexplore.ieee.org.ezproxy.fau.edu/stamp/stamp.jsp?tp=&arnumber=5076651&isnumber=5076603

[19] Luthria, H.; Rabhi, F.; Briers, M.; , "Investigating the Potential of Service Oriented Architectures to Realize Dynamic Capabilities," Asia-Pacific Service Computing Conference, The 2nd IEEE , vol., no., pp.390-397, 11-14 Dec. 2007doi: 10.1109/APSCC.2007.70 URL: http://ieeexplore.ieee.org.ezproxy.fau.edu/stamp/stamp.jsp?tp=&arnumber=4414486&isnumber=4414426

[20] Cuadrado, F.; Garcia, B.; Dueas, J.C.; Parada, H.A.; , "A Case Study on Software Evolution towards Service-Oriented Architecture," Advanced Information Networking and Applications - Workshops, 2008. AINAW 2008. 22nd International Conference on , vol., no., pp.1399-1404, 25-28 March 2008 doi: 10.1109/WAINA.2008.296 URL: http://ieeexplore.ieee.org.ezproxy.fau.edu/stamp/stamp.jsp?tp=&arnumber=4483115&isnumber=44828