Design and Implementation of a Web Service Application for SCM

A project initial report submitted to The University of Manchester for the degree of
MSc Software Engineering
in the Faculty of Engineering and Physical Sciences

2009

ZHANGMIN LU
SCHOOL OF COMPUTER SCIENCE

Table of Content

Abstract

From face-to-face communication to paper-based communication to electronic communication to global communication via Internet, the cost of communication has been decreasing dramatically. Consequently, the productivity and efficiency of human activities are being improved continuously. In the Supply Chain Management (SCM) world, Electronic Data Interchange (EDI) has been adopted by the large enterprises for business transactions such as placing an order, sending despatch advice, and receiving invoice. However, the cost and effort required to deploy traditional EDI is significant for small and medium-size enterprises, and even some large enterprises which have many divisions running different legacy information systems on heterogeneous platforms.

Fortunately, new standards and methodologies have been developed in the Information and Communication Technology (ICT) field to provide better business solutions. Among these emerging techniques, Web services, Electronic Business eXtensible Markup Language (ebXML), and Universal Business Language (UBL) are the primary ones for e-Business integration. The conventional SOAP-based Web services technology mainly focus on exposing the business operations as remote procedure calls (RPCs); this approach is complex and expensive, and requires considerable amount of work on the customisation of invocation methods. Therefore, this project will investigate another lightweight and resource-oriented alternative, the REpresentational State Transfer (REST) style to implement supply chain activities as a set of Web services.

While Web services establish the means of communication, ebXML is the framework for e-Business integration and collaboration including business registries, processes, profiles, and agreements. In addition, UBL is a universal XML library designed for business document exchange. With these advanced technologies, this project aims to develop a prototype using resource-oriented REST-based Web services, ebXML e-Business framework, and UBL standard business documents for SCM, thus demonstrating the benefits of applying and integrating modern techniques into business scenarios.

Chapter 1 Introduction

This chapter gives an overview of the research areas this project is interested in, and then outlines the scope and context of the investigation. After that, it introduces the problems to be solved.

1.1 Overview

The Internet has become the most effective world-wide communication channel. The World Wide Web is the most common use over the Internet, people use web sites to buy books and CDs (Amazon), retrieve information (Wikipedia), build up social network (Facebook, MySpace), search other web sites (Google, Yahoo!), buy and sell their merchandise (eBay), and so on.

In the trading world, the new business model over Internet is the so-called Electronic Business (e-Business). This evolutionary e-Business model covers a wide span and focuses on streamlining the entire value chain network including supply chain, automating business transactions, and collaboration between business partners. The main task of e-Business integration is automation of the commercial and administrative operations between partners within a value chain network.

Web services are the means of interoperating between different computer applications running on diverse platforms or environments. The objective of Web services is to provide interfaces for distributed systems to communicate with each other, and Web services utilise standards such as HTTP, URI (Universal Resource Identifier), and XML to share and exchange information via Internet. By constructing and deploying Web services in the distributed computing environment, applications components can be reusable and existing software can be connected seamlessly.

Business registries, processes, profiles, and agreements are other important elements of e-Business collaboration. Business registry is a repository service where organisations can register and search for Web services like yellow pages for telephone numbers. Business processes services are used to define and promote a reference model of business information exchange. Business profiles and agreements services provide a means for business partners to interact through their technical capabilities.

Document exchange is at the centre of e-Business transactions. XML has claimed to have solved the standard for document exchange; however, with XML only one cannot resolve the issue of semantics of the documents. As a result, there is a demand for a standard business vocabulary to tackle the semantic conflicts of the data in different situations or contexts, and define conceptual models for the semantics of business documents using XML syntax.

Supply Chain Management (SCM) is the cross-discipline management of a network of interconnected business processes, which has been in existence and evolving for

several decades. There is a public recognition that supply chain management is one crucial area that affects every business irrespective of size, sector or location ranging over suppliers, customers, and business partners could work collaboratively to deliver goods or services as required. In this sense, a common cross-industry standard reference model of supply chain management would help to address and improve supply chain best practices, and enhance the communication between cooperative supply chain parties.

1.2 Outline of the Scope and Context of Investigation

Currently, there are two approaches for implementing Web services, one is the traditional SOAP-based Web services, and the other is the emerging REST-based Web services. The SOAP-based Web services encapsulate business operations as remote procedure calls (RPCs) into the SOAP message envelope, whereas the REST-based Web services just simply use standard protocol HTTP, and universal naming and addressing scheme URIs to expose and consume the services. Generally speaking, the REST-based approach is a simple and cost-effective way to implement Web services.

Electronic Business eXtensible Markup Language (ebXML) is an e-Business framework that includes a suite of standards such as ebXML Registry/Repository for storing and sharing business registries, ebXML Business Process (ebBP) for designing business process models, ebXML Collaboration Protocol Profile and Agreement (ebCPPA) for setting up an individual entity's technical capabilities and establishing the contract between entities.

Universal Business Language (UBL) is the first International effort and specification for e-Business document exchange. It consists of a royalty-free XML library of standard business documents such as purchase order and invoice. UBL directly supports the existing business by eliminating the data re-entering in current paper-based supply chains. More important, UBL provides standard business vocabulary for partners to communicate base on common understanding of the semantics of documents.

Supply-Chain Operations Reference (SCOR) model is a global industrial initiative in the supply chain domain. Leading by the International non-profit consortium Supply-Chain Council and embracing by thousands of organisations, SCOR establishes a common reference model to manage supply chains processes and improvements.

This project will investigate and evaluate relevant technologies above, and then implement a prototype by integrating REST-based Web services, ebXML framework, and UBL business document standard in the context of a small fictional supply chain management process. The SCOR model is used as reference in the implementation. The participants include a PC manufacturer company ABC, several PC components suppliers, and ABC's third party warehouse operator.

1.3 Definition of the Problem

1.3.1 The need for implementing SCM as Web services

Advances in ICT have lead to fundamental changes in the way companies conduct and manage their businesses. With the sophisticated solutions like ERP (Enterprise Resource Planning) and WMS (Warehouse Management System), the efficiency of internal operations has been improved greatly. However, these solutions do not help in the automation, integration, and collaboration for the entire supply chain. With the fast-paced and ever changing global market, mass customisation, agile product development, and extensive outsourcing, companies are experiencing more and more pressure on their supply chains. The need for implementing supply chain management processes as Web services is imperative to cut cost, and raise competitiveness of the enterprise, its suppliers, and business partners.

1.3.2 EDI and traditional Web services are too expensive

EDI technology has been developed and adopted for many years. However, while about 95% of Fortune 1000 enterprises are using EDI, there are only 2% of small and medium-size enterprises that can afford to use it [1]. Thus, EDI has been commonly recognised as an expensive and complex solution because of the obscure syntax and the difficult supporting applications, the high cost of implementation, and the rigid structure.

Similarly, the traditional SOAP-RPC approach for Web services implementation has limitations. Firstly, the SOAP message format is quite long-winded, so the processing of SOAP messages can be very slow when transmitting large documents. Secondly, when using SOAP-based Web services, the address of target resources and business operations are hidden in the SOAP message envelope, thus requiring an extra server to parse and check the SOAP message to process the service request. At last, different suppliers may have different interfaces for the same operation e.g., getOrder, getOrderInfo, or retrieveOrder for a review order function. The expense required to design and program this type of application is still significant.

1.3.3 A Resource modelling technique for REST-base Web services

Simple is beautiful, something like CRUD (Create, Read, Update, and Delete) is a major part of nearly every computer software. There is one example that has proved the power of few methods and that is SQL, the famous SELECT, UPDATE, INSERT, and DELETE can manipulate the database effectively. Similar to SQL, HTTP, the standard protocol used by REST-based Web services also has uniform interfaces GET, POST, PUT, and DELETE [2]. However, developing REST-based Web services to support and automate business activities requires sophisticated system analysis and design. In particular, in order to expose the application capabilities through resources with uniform interfaces, a ‘resource modelling’ technique is required which can help to program the resource methods that fulfil the required capabilities.

Chapter 2 Background

The background chapter discusses and evaluates some of the current contemporary technologies. Firstly, two approaches for Web services are outlined, including a traditional SOAP-RPC based one, and another alternative called RESTful Web services [3]. Secondly, the ebXML framework comprising of a set of standards for e-Business integration and collaboration, is described. Thirdly, the business document exchange specification UBL is presented. Finally, the Supply-Chain Operations Reference-model (SCOR) is introduced.

2.1 Web services

2.1.1 REST vs. SOAP

The current generation of Web services are built on SOAP with the intention of being an interoperable protocol in the Internet world equivalent to the DCOM (Distributed Component Object Model) or CORBA (Common Object Requesting Broker Architecture) in the inter-application world. However, these technologies heavily rely on the RPC model which is only suitable for the closed-world situation. SOAP is designed to encapsulate RPC functionality into the SOAP envelope message, but the contents of elements ‘Header’ and ‘Body’ within the envelope message are application dependent and not defined in the SOAP specification [4] [5]. Therefore, when applying SOAP in an open multi-user environment with no global understanding of semantics of RPCs, sharing and changing interfaces of RPC model is extremely difficult and costly because the clients, servers, and the intermediaries are application specific and tight-coupled.

An example of a typical SOAP Web services request and response is shown in Example 2-1. The request function ‘getinfo’ is application defined and encapsulated in the SOAP message, and the response information is also embedded in the SOAP body.

Example 2-1 SOAP Web services request/response sample

Request:

```
POST /temp HTTP/1.1
Host: www.harddisk.com
Content-Type: application/soap+xml

<?xml version="1.0"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
    <env:Body>
        <m:getinfo   env:encodingStyle="http://www.w3.org/2003/05/soap-encoding"
            xmlns:m="http://www.harddisk.com/">
            <m:partnumber>HD-150G</m:partnumber>
        </m:getinfo>
    </env:Body>
```

```
</env:Envelope>
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml

<?xml version="1.0"?>
<env:Envelope xmlns:env=http://www.w3.org/2003/05/soap-envelope>
    <env:Body>
        <m:getinfoResponse env:encodingStyle="http://www.w3.org/2003/05/soap-encoding"
            xmlns:m="http://www.harddisk.com/">
            <Name>150G Harddisk</Name>
            <Description>This part is 150GB 7200RPM SATA Harddisk</Description>
            <UnitCost currency="GBP">15</UnitCost>
        </m:getinfoResponse>
    </env:Body>
</env:Envelope>
```

As it can be seen from Example 2-1, the SOAP message is quite tedious, and the URI used in SOAP-based approach is not to identify the target resource, but rather to point to a SOAP server. Therefore, it is hard, and less likely, for a Web server to determine which resource is actually being requested. An additional SOAP server is required to look inside the SOAP message to check which resource is being targeted as in Figure 2-1. Because of the verbose SOAP message format and the extra server layer, SOAP can be considerably slower than other technologies.

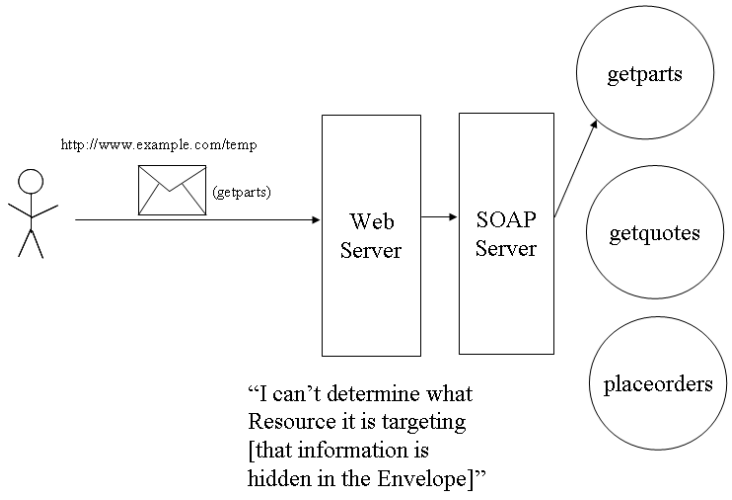

Figure 2-1 SOAP request to servers

Hence, RESTful Web services is getting popular for its simplicity and generic interfaces – HTTP GET, POST, PUT, and DELETE. REST is a network-based architectural style introduced by Dr. Roy Fielding in his PhD dissertation in 2000 which outlines several design constraints behind the modern Web architecture [6]. In

the beginning, REST derives the constraints from other application styles such as: (1) client-server constraints focus on separation of concerns principle and allow the clients and servers to evolve independently, (2) client-server communication must be stateless thus easing the task of monitoring and recovering from failures because each request contains all the necessary information, (3) cache constraint is used to improve efficiency, scalability, and performance by eliminating some interactions through cacheable data, (4) layered system constraints allow an architecture to have hierarchical layers that improve system scalability by balancing of services across multiple networks [6].

The additional constraint distinguishes the REST style from other styles and is the uniform interface between components, which emphasises the generality of component interface. This constraint induces the other four interface constraints including resource identifiers, manipulation of a resource by using a representation, self-descriptive messages, and hypermedia engine of application state [6]. Example 2-2 shows a RESTful Web services request and response. The request is just a simple HTTP GET method, and the response is just a plain XML document.

Example 2-2 RESTful Web services request/response sample

Request:

```
GET /parts/HD-150G HTTP/1.1
Host: www.harddisk.com
Accept: application/xml
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/xml

<?xml version="1.0"?>
<part>
    <Name>150G Harddisk</Name>
    <Description>This part is 150GB 7200RPM SATA Harddisk</Description>
    <UnitCost currency="GBP">15</UnitCost>
</part>
```

As shown in Example 2-2, in the RESTful Web services, the URI identifies the resource that is desired (e.g., orders), so a Web server can easily decide, based upon the identified resource as in Figure 2-2. Furthermore, for RESTful Web services with explicit URI, the system/network administrators can apply Access Control Lists (ACLs) to secure services because each of them has a specific URI. Moreover, the HTTP method used in RESTful Web services enables the administrators to protect the services by enforcing the access control policy like disabling certain operations (e.g., PUT or DELETE). The dotted lines in Figure 2-2 give the example of access control for RESTful Web services.

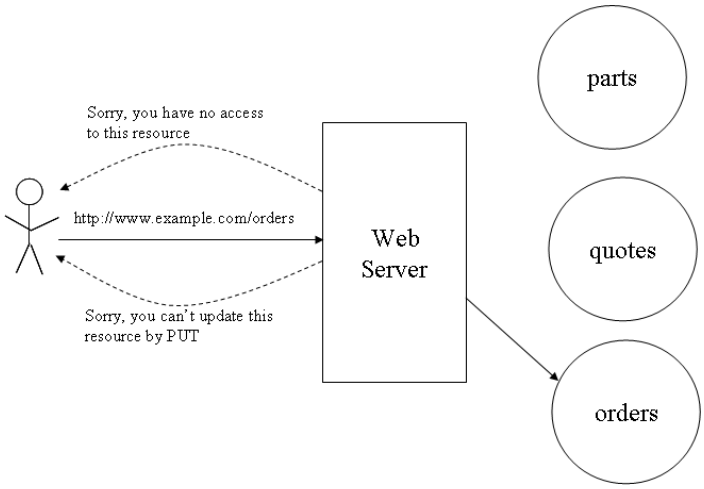

Figure 2-2 RESTful Web services request to server

2.1.2 Resource Modelling for REST

The advantages of RESTful Web services are that they are lightweight, human readable, easy to build, and trivial to test using a browser or common tools such as curl and wget. However, REST is not a silver bullet, and REST emphasises a software design discipline of many URIs and few methods [8]. Hence, the challenge to RESTful Web services is to find an effective mechanism to map business logics to resources and limited uniform interfaces. This requires that application designers take a fundamental shift to rethink the problems in terms of manipulations of addressable resources rather than the traditional object-oriented way about objects and functions [8] [9]. This may be the most difficult part to implement for the programmers who are familiar with the RPC approach of integrating existing applications through unrestricted user-defined APIs.

Therefore, this project will investigate a resource modelling technique to help practitioners of RESTful Web services build up the application's resource model. Since it is the application's responsibility to understand the format of information (i.e. content types), this project will focus on the research of resource modelling on the identifier of the resource, and the action required for an application to interact with a resource [10]. The data flow modelling technique used in SSADM (Structured Systems Analysis and Design Method) provides a reference model for the resource modelling development, because it shows the processing needed in an information system to maintain (create, read, update, and delete) the data.

A data flow model is composed of processes, external entities, and data stores. Each data flow from an external entity to a process can represent one of two operations: (1) an event with data items to perform an update on the data, (2) a triggering of an enquiry on the data [11]. These operations can be mapped to the HTTP GET, POST, PUT, and DELETE methods based on the nature of the processes and links between processes and data stores. In addition, the data stores in the data flow model are the basis for constructing URIs of the resources in RESTful Web services.

2.2 ebXML – The e-Business Framework

Electronic Business is a buzzword in the ICT industry which covers all aspects of the value chain. Electronic Business using eXtensible Markup Language (ebXML) is a joint initiative by OASIS (Organization for the Advancement of Structured Information Standards) and UN/CEFACT (United Nations/ECE agency CEFACT) to enable the enterprises of any size to conduct businesses over the Internet; it defines a family of XML-based specifications to support global use of electronic business transactions in an interoperable manner by all trading partners [12].

Firstly, the ebXML Registry/Repository is a platform independent technology for storing and sharing of business artefacts, discovering businesses, and facilitating business exchanges via collaboration profiles [13]. To use an informal analogy, the ebXML Registry/Repository is to digital artefacts, and the library is to books. While the registry is analogous to the index of books in the library, the repository is analogous to the book shelves in the library [14] [15]. For Web services, ebXML Registry/Repository can be used as a database to publish and discover services, thus enabling the cooperation across organizational boundaries. ISO has approved the ebXML Registry as an ISO standard under name of ISO/TS 15000, parts 3 and 4 [16].

Secondly, ebBP (ebXML Business Process), previously called the ebXML Business Process Specification Schema (ebBPSS), is an OASIS technical specification for designing collaborative business processes [17]. It defines a standard process definition language to configure business systems to support the collaboration between partners rather than the processing accomplished within one individual business entity [18]. An ebBP definition describes interoperable business processes that allow collaborating parties to cooperate and achieve a shared business objective, which contains business transaction, business signals exchanged in a business transaction, and choreography of business transaction(s) that comprise business collaboration [19]. Web services software components can be specified to work with ebBP technical specification to execute business collaborations effectively.

Finally, the ebXML Collaboration Protocol Profile and Agreement (ebCPPA, aka ISO/TS 15000-1) standard provides two sets of information for business collaborations. One set – CPP (Collaboration Protocol Profile) expresses data about a single business partner's technology capabilities such as delivery channels and transport bindings to engage in e-Business collaborations with other partners. The second set – CPA (Collaboration Protocol Agreement) defines the capabilities that two parties have agreed bilaterally to use in the business collaboration protocols [20]. However, CPA is designed to be a technical agreement, not a legal contract.

An overview of the OASIS ebXML e-Business framework is shown in Figure 2-3. The first step is design the business process model for a specific industry. Then, the second step is about implementing and registering the profiles (CPPs). After reaching the agreement (CPA), the partners can exchange business documents.

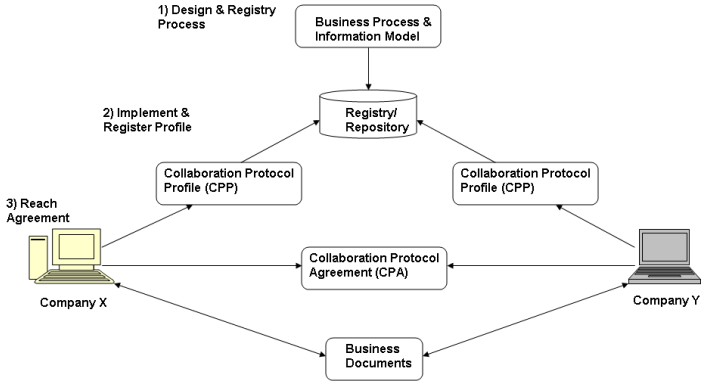

Figure 2-3 OASIS ebXML framework Overview

2.3 Standard Business Vocabulary – Universal Business Language (UBL)

Documents are widely used in business transactions such as orders and invoices. Every document has its purpose and contains adequate information for a particular transaction. It provides the interfaces for people and interfaces to business processes. XML has been accepted as the standard mechanism for document exchange. But, in order to conduct business, partners need to understand the semantics of the documents. Universal Business Language (UBL) is the language to capture business information and abstract common patterns of business documents. It is an international initiative to define an open royalty-free standard XML documents library of e-Business which is developed by an experienced and accountable OASIS technical committee with a variety of industry participants [21].

The OASIS UBL specification is intended to meet the requirements of industries by defining a generic extensible XML vocabulary for business documents interchange. Specifically, UBL consists of the following elements: (1) a library of XML schemas for frequently-used data components of everyday business documents such as ‘Address’, ‘Item’, and ‘Price’, (2) a set of XML schemas that are composed by the UBL library components for common business documents such as ‘Order’, ‘Despatch Advice’, and ‘Invoice’ which can be used in generic supply chain context [22]. UBL is the first international implementation which is compliant with the ebXML Core Components Technical Specification (i.e. ISO/TS 15000-5). UN/CEFACT has recognised UBL as the first-generation XML documents for e-Business collaboration which means UBL can be safely adopted for now and into the future [21].

Example 2-3 is a simple UBL Quotation document. Two important concepts here are the ‘cbc’ (Common Basic Components) and ‘cac’ (Common Aggregate Components) schemas. The former defines the Basic Business Information Entities (BBIEs) that are used as ‘leaf nodes’ throughout UBL, and the latter defines the Aggregate Business Information Entities (ABIEs) that are used to construct the main documents [22].

Example 2-3 OASIS UBL Quotation document fragment sample

```
<cac:LineItem>
    <cbc:ID>1</cbc:ID>
```

```
    <cbc:Quantity unitCode="KG">100</cbc:Quantity>
    <cac:Price>
        <cbc:PriceAmount currencyID="GBP">100.00</cbc:PriceAmount>
        <cbc:BaseQuantity unitCode="KG">1</cbc:BaseQuantity>
    </cac:Price>
    <cac:Item>
        <cbc:Name>OliverOil</cbc:Name>
        <cac:BuyersItemIdentification>
            <cbc:ID>020010N0674</cbc:ID>
        </cac:BuyersItemIdentification>
        <cac:SellersItemIdentification>
            <cbc:ID>2-010-N0674</cbc:ID>
        </cac:SellersItemIdentification>
    </cac:Item>
</cac:LineItem>
```

2.4 The SCOR Model

Supply-Chain Operations Reference-model (i.e. SCOR) is a de facto standard reference model for supply chain management across industry boundaries [23]. It is developed by Supply-Chain Council to establish a unique framework to define business process into a unified structure to support the communication between business partners and to improve the productivity of the entire supply chain. SCOR has been designed to use common process building blocks to describe simple or complex supply chain activities, thus enabling all interested parties to be linked to describe the operations of virtually any supply chain [24]. Table 2-1 shows some components in the three levels of SCOR model.

Level	***Description***	***Components***
1 – Top Level	Process Types	Plan, Source, Make, Delivery, Return
2 – Configuration Level	Process Categories	D1 – Delivery Stocked Product D2 – Delivery Make-to-Order Product D3 – Delivery Engineer-to-Order Product
3 – Process Element Level	Decompose Processes	D1.1: Process Inquiry & Quote D1.2: Receive, Enter & Validate Order D1.11: Load Product & Generate Shipping Docs D1.13: Receive & Verify Product by Customer D1.15: Invoice

Table 2-1 Supply-Chain Operations Reference-model

2.5 Summary

This chapter described the background of relevant technologies including Web services, ebXML e-Business framework, and the XML-based business language UBL, and the SCOR model. The next chapter presents the research methodology and approach used in this project.

Chapter 3 Research Methods

This chapter outlines the significance and aims of the project at first. Then, the 'constructive research' and 'software prototyping' methodologies are discussed in section 3.3. The research approach is presented in section 3.4 followed by the expected results. The tools used in this project are described in section 3.6. This chapter closes with the limitations and project plan section in 3.7 and 3.8 respectively.

3.1 Significance of the project

Instead of following the traditional SOAP-RPC approach, this project will explore the lightweight RESTful Web services based on HTTP+URI+XML, and open international standards ebXML e-Business framework and UBL business documents to make partner-facing Web services simple and affordable for enterprises of any size in supply chain domain. This combined solution has not been found in the initial literature review; and it is worth investigating and testing.

In a SOAP world, business functions are encapsulated in the SOAP message envelope, and different suppliers may have different interfaces for one function e.g., placing order, thus requiring different SOAP clients to be implemented. Furthermore, since the actual information is encapsulated in the payload of the SOAP message, an additional server layer is required to parse the SOAP message to transfer or route the message. Because of the development workload of different clients, verbose SOAP message format, and extra server layer, the SOAP-based approach becomes heavyweight and costly. On the other hand, in a REST world, different suppliers will have different URIs for placing order, but using varied URIs in REST client is quite a simple procedure by sending the plain purchase order XML document to different URIs via a single HTTP POST method.

In addition, this project will investigate a new resource modelling methodology similar to data flow modelling to identify a resource and effectively convert traditional operations into a few simple verbs. This will support the job of RESTful Web services developers to design the resource model, and then expose and consume the services effectively.

3.2 Aims of the project

Chapter 1 showed the need for implementing supply chain management processes as Web services, and chapter 2 discussed the background of relevant technologies. The aims of this project are:

- Analysis of RESTful Web services for supply chain activities, and relevant technologies – ebXML specifications and UBL business documents.
- Implementation of a lightweight and integrated solution for supply chain management using RESTful Web services, ebXML framework, and UBL.
- Research of the resource modelling technique for RESTful Web services base on data flow modelling.

3.3 Research Methodology

The constructive research methodology will be adopted for this project. The constructive research approach is a typical research procedure in computer science field for designing, implementing, and evaluating a software construction, intended to solve domain problems faced in real world, and by that means, to create knowledge and make a contribution to the theory of discipline [25]. The constructive research methodology includes practical relevance and theoretical relevance, the former indicates that the construction should produce a solution for a domain problem in practice; the latter indicates that the research should generate some related theoretical knowledge to refine a theory, even develop a new one. By using the constructive research methodology, this project will analyse, design, and implement an integrated solution for supply chain management with lightweight RESTful Web services, and the open royalty-free standards ebXML and UBL. Meanwhile, the project is investigating a novel resource modelling technique as an improvement for RESTful Web services development.

This project will use another common methodology in computer science – software prototyping to implement the solution. A software prototype is different from a full-scale software system; it is just a partial implementation of a system with the purpose of exploring the problem to be solved or testing a solution to that problem. In particular, the experimental prototyping approach will be used for this project to verify a possible solution to the domain problem [26]. In experimental prototyping, the basic functionality is understood but some of design issues need to be resolved. An experimental prototype can implement any subset of the target system. This project will only focus on several process elements in the delivery process of the entire supply chain operations reference model.

3.4 Research Approach

The research approach of this project is divided into three phases including analysis, design, and implementation.

3.4.1 Analysis

The research of RESTful Web services approach for supply chain management forms the basis of the analysis phase. The REST architectural style for Web services development and the SCOR process reference model for supply chain management are being analysed to suggest a model in which individual data sources and services can be tied together on demand to generate a desired result with minimal effort. Orchestrating a different configuration with new processes or partners will not be costly and complicated. Every raw material in the system (i.e., data) can be identified as a resource and referenced via the identifier of resource. By analysing the concepts of modern Web architecture described in Dr. Roy Fielding's thesis [6], it is sensible for enterprises to implement and expose URI-addressable RESTful Web services to external partners, thus supporting sophisticated supply chain collaboration across organisational boundaries. Figure 3-1 shows the proposed business activity diagram.

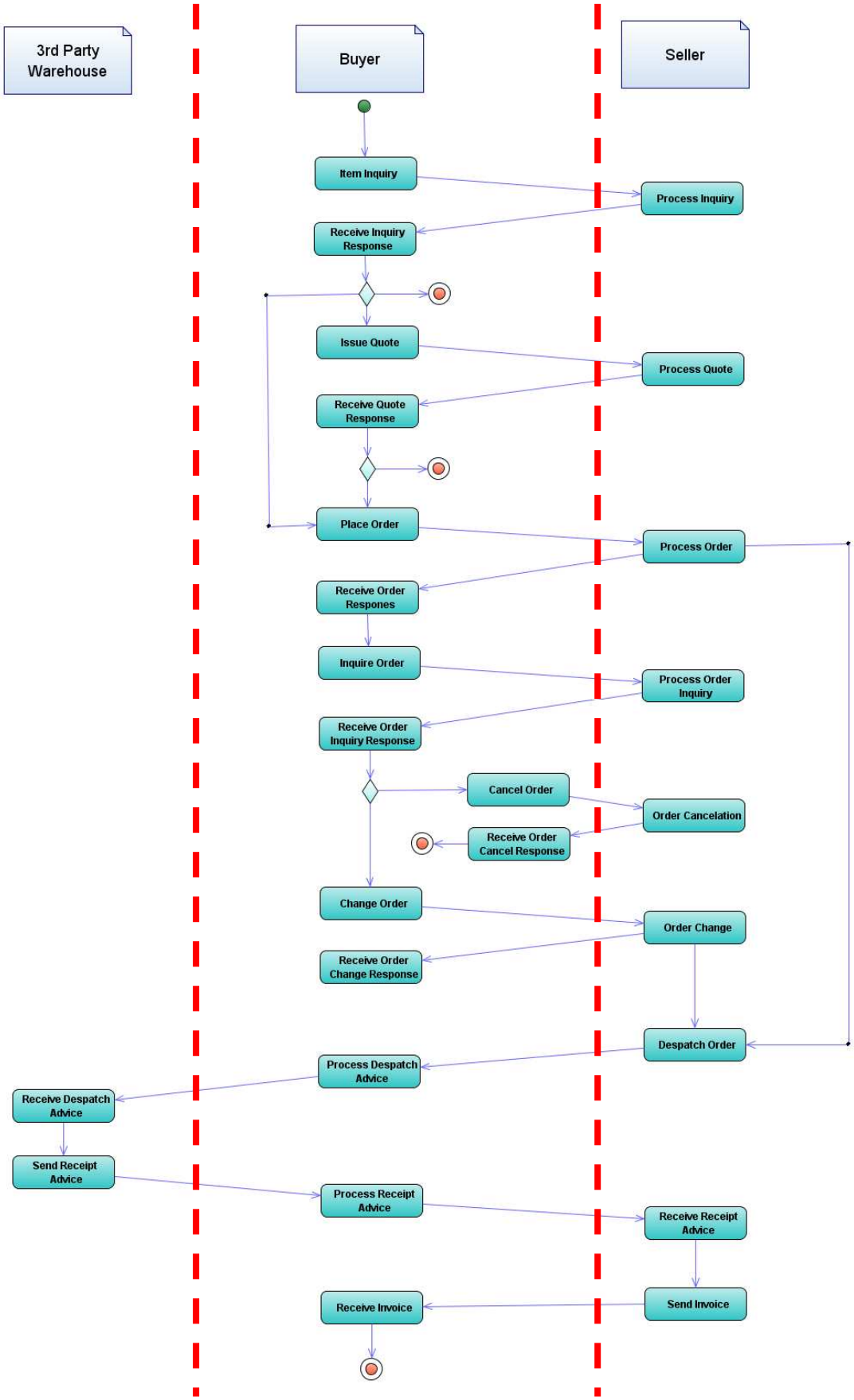

Figure 3-1 Proposed Business Activity Diagram

Particular analysis will be performed on the resource modelling technique for RESTful Web services development. In a RESTful application, resource is the key abstraction of information, and the uniform interface between components is the unique feature [6]. The challenge for a RESTful Web services design is to model the business processes around resources and few methods. Therefore, this project will analyse the data flow modelling, which is a technique for graphically representing the processes on data stores including creating a new an entity (add), retrieving an entity (read), updating an entity (change or modify), and deleting an entity.

Figure 3-2 and Figure 3-3 show two samples, one simple data flow diagram (DFD) and one proposed resource model for RESTful Web services. The place order process in DFD is used to create a new order, and can be mapped to a HTTP POST action; the review order process in DFD is used to retrieve order information, and can be mapped to a HTTP GET action.

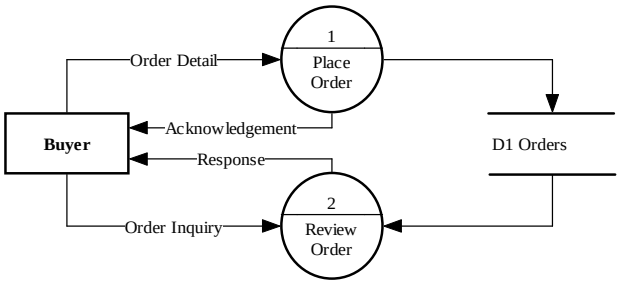

Figure 3-2 Sample Data Flow Diagram (DFD)

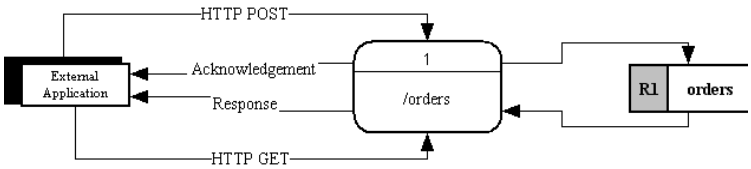

Figure 3-3 Proposed Resource Model for RESTful Web services

Ancillary analysis will be conducted on ebXML framework for e-Business and UBL specification. Within the ebXML framework, the ebXML Registry/Repository standard, the ebBP Business Process specification, and the ebCPPA Collaboration Protocol Profile and Agreement standard will be studied. For UBL, the theoretical rationale will be analysed in general, and the quotation and order to invoice processes will be investigated in detail for the implementation of this project.

3.4.2 Design

The design phase transforms the requirements gathered during the analysis phase into a computer system design, which details what the system is required to do [11]. This project will be decomposed into a Web services design task, a business process modelling task, a collaboration profile and agreement design task, and a business document engineering task. The SSADM method will be adopted, plus the UML will be used to design the core classes and their associations for the prototype system.

3.4.3 Implementation

The system design provides the architecture, components, modules, interfaces, and data for building and testing the new system. It is during the implementation phase, the design is converted into computing source code as a program, software component, or other computer system. This project will deliver following main components:

- A prototype of RESTful Web services for supply chain management, this includes: (1) a set of Web services for the seller (supplier) to process the item inquiry, respond to quote, receive new order, process order e.g., inquiry, cancellation or change, and receive receipt advice from the buyer (customer), (2) a couple of Web services for the buyer to process despatch advice from seller, process receipt advice from third party warehouse operator, and receive invoice from seller, (3) a Web service for the third party warehouse operator to receive despatch advice from buyer.
- A prototype of RESTful clients, which includes: (1) a client for the buyer to inquire item information, issue quote, place order, review, change or cancel order, (2) a client for the seller to despatch order, and send invoice, all sellers will share the same client template with only the seller's name being different, (3) a client for the third party warehouse to process and send receipt advice.
- An ebBP business process document (diagram) to describe the supply chain activities involved in this project. It will cover the interactions between buyer, seller, and third party warehouse operator.
- One CPP for the buyer, one CPP for the third party warehouse operator, several CPPs for the sellers, and a couple of CPAs will be constructed according to the corresponding interfaces every role has.
- Some selected UBL business documents will be tailored and used for the business transactions such as order, despatch advice and invoice.
- An ebXML Registry/Repository server will be implemented to provide the storage for sharing business process document, collaboration profiles and agreements, and relevant document schemas.

3.5 Expected Results

The analysis, design, and implementation of this project will be used to test the following expected results:

- It is possible to use the simple and lightweight RESTful Web services to implement a sophisticated supply chain solution.
- It is possible to derive a resource modelling technique for RESTful Web services development from the data flow modelling technique.
- It is possible to apply the Universal Business Language (UBL) as a common business vocabulary for the supply chain interactions.

3.6 Project Tools

In order to enable other interested researchers or parties to replicate or use the work of this project, all the software development tools chosen for the implementation are licensed under either free software licence or open source licence as in Table 3-1.

Tool name	*Description*	*License*
Canonical Ubuntu GNU/Linux	Computer Operating System used for this project	GNU General Public License and other licenses
Ruby	Cross-platform, general purpose interpreted and object-oriented language	Ruby License, GNU General Public License
Ruby on Rails	Open source Web application framework	MIT License (Free software license)
MySQL	Popular RDBMS for web applications	GNU General Public License
Freebxml ebBP Editor	Design generic or specific business process specification	freebxml License, Open Source Software,
Freebxml ebCPP Editor	Create ebXML Collaboration Protocol Profile (CPP)	freebxml License, Open Source Software
Freebxml Registry	Free reference implementation of ebXML Registry standard	freebxml License, Open Source Software

Table 3-1 Project Software Development Tools

A laptop computer with duo 1.8GHz CPUs and 1 GB of RAM is used to carry out the development of this project. However, the prototypes implemented by this project should work on an older hardware configuration which meets the minimum requirement of the software development tools. Although there are three roles – buyer, seller, and third party warehouse operator, it is sufficient to use one computer to host the server and clients together.

3.7 Limitations

This project has following limitations:

- The scope of this project will only cover a few process elements of one subset of delivery process in the SCOR model. The implementation of complete SCOR model would be more complex.
- The research of resource modelling technique will focus on the identifier of resource and application interfaces. And use XML as the sole representation.
- This project will just use the public samples of CPP, CPA, and UBL from the OASIS web sites as the templates with slightly modification to test various scenarios in the project.
- The basic security/authentication will be adopted, while in real world, more sophisticated security/authentication mechanism is enforced.

3.8 Project Plan

The project plan is an important tool for documenting planning assumptions and decisions, facilitating communication among stakeholders, and is served as schedule baselines. Figure 3-4 shows the plan of this project, which consists of three main stages including analysis and design, implementation, and writing up dissertation. The exams review period is included in the plan for clarifying the time gap only.

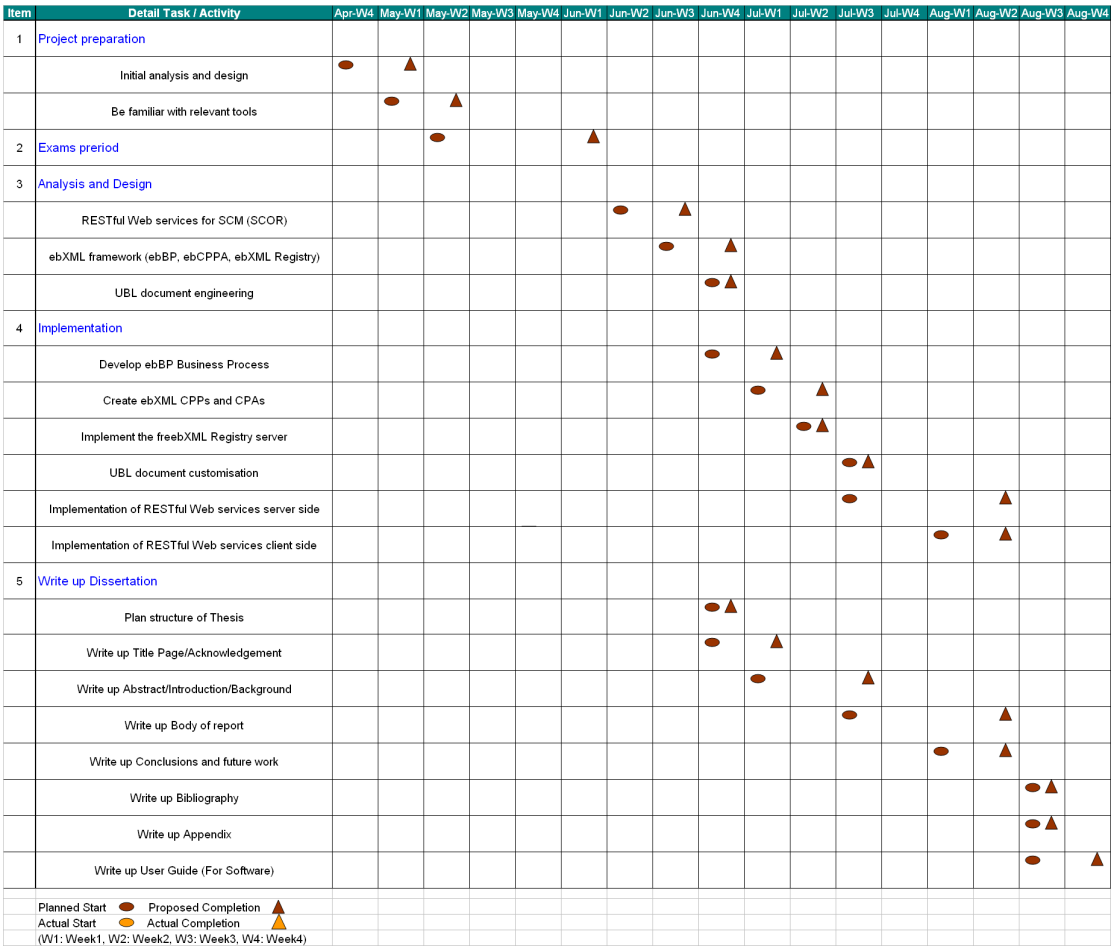

Figure 3-4 Project Plan

3.9 Summary

This chapter showed the project aims, the research methodology and research approach of the project. The expected results were presented, followed by all software tools and the hardware environment used in this project. This chapter closed with limitations of the project, and the project plan for this research.

Chapter 4 Conclusion

The conclusion chapter briefly conducts a critical analysis and evaluation on the technologies used in this project. Then it summarises the achievements and improvements, followed by the discussion of future work.

4.1 Critical analysis and evaluation

RESTful Web services, ebXML framework, and UBL are the technologies used in this project. Among them, ebXML and UBL are both International standards for e-Business recommended by OASIS and UN/CEFACT. The core technology will be evaluated here is the RESTful Web services, or simply REST.

REST is not a silver bullet, it has strengths and weaknesses. The strengths of REST include: (1) simplicity, because the uniform interface is invariant and there is no problem of breaking clients, (2) scalability, because the separation of concerns principle allows the client and server to evolve independently, (3) visibility, because the communication is stateless, and a system does not need to look beyond a single request to understand it, (4) interoperability, because REST advocates standards make today's Web successful like HTTP, URI, HTML, and XML, (5) REST conforms the Axioms of Web architecture [27]. The weaknesses of REST include the challenge to identify and model the resources appropriately in the Web applications, and informal semantics and syntax are just mainly for human readers to use.

Likewise, the adoption of RESTful Web services has opportunities and threats. Due to the complexity of SOAP, 85% of Amazon's Web services usage was of the REST interface in 2003 [28], and Google deprecated its SOAP Search API in 2006 [29]. The low entry barriers and grassroots approach give the opportunity to RESTful Web services to thrive on the Web. However, commercial hype over SOAP and lack of support and tools for REST mean REST is still largely unknown to enterprises.

4.2 Achievements and Improvements

This project is expecting an integrated e-Business solution for supply chain management with RESTful Web services, ebXML framework, and UBL business document standard including a set of RESTful Web services for supply chain management process, a prototype of REST client to utilise the services, an ebBP business process model base on the SCOR model, a couple of ebCPPA profiles and agreements, an ebXML Registry/Repository server, and UBL business documents for supply chain interactions.

In the meantime, a pragmatic resource modelling technique is being investigated as supplementary improvement for RESTful Web services design. The data flow modelling technique which is to maintain (create, read, update, and delete) an information system will be used as reference model for the design of resource modelling technique.

4.3 Future Work

The current approach requires a certain amount of human agreement for services construction and consumption. Therefore, Simple Knowledge Organization System (SKOS) [30] built on Resource Description Framework (RDF) [31] is worth researching to allow automatic registration and discovery of Web services. Unlike the formal knowledge representation language – Web Ontology Language (OWL) [32], SKOS provides a simpler standard, low-cost approach for existing knowledge representations such as thesauri, classification schemes, and taxonomies to be used in the Semantic Web. Furthermore, SKOS defines a family of lightweight languages for developing and sharing new concept schemas (e.g., taxonomies, structured controlled vocabularies, etc.), thus enabling data sharing across a variety of applications [30].

4.4 Summary

This final chapter of the project initial report evaluated RESTful Web services at first. Then it discussed the achievements, improvements and future work for this project.

References

[1] Waldt, D. (2003). ebXML Web Services & EDI. http://www.ebxml.org/presentations/ws_ebxml_london_may03.pdf. XML Europe 2003, London.

[2] Prescod, P. (2002). Second Generation Web Services. http://webservices.xml.com/pub/a/ws/2002/02/06/rest.html

[3] Richardson, L. and S. Ruby (2007). RESTful web services. Sebastopol, California, O'Reilly.

[4] W3C Recommendation (2007). SOAP Version 1.2 Part 0: Primer (Second Edition). http://www.w3.org/TR/2007/REC-soap12-part0-20070427/

[5] W3C Recommendation (2007). SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). http://www.w3.org/TR/2007/REC-soap12-part1-20070427/

[6] Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation, University of California, Irvine, http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

[7] Tilkov, S. (2008). Addressing Doubts about REST. http://www.infoq.com/articles/tilkov-rest-doubts

[8] Prescod, P. (2002). REST and the Real World. http://webservices.xml.com/pub/a/ws/2002/02/20/rest.html

[9] Manes, A. T. (2008). REST is about Resources. http://apsblog.burtongroup.com/2008/03/rest-is-about-r.html

[10] Wikipedia (Accessed on 09/04/2009). Representational State Transfer. http://en.wikipedia.org/wiki/Representational_State_Transfer

[11] Weaver, P. L., N. Lambrou, et al. (2002). Practical business systems development using SSADM : a complete tutorial guide. Harlow, Financial Times Prentice Hall.

[12] OASIS ebXML.org (Accessed on 10/04/2009). About ebXML. http://www.ebxml.org/geninfo.htm

[13] Chappell, D. A. (2001). Professional EbXML foundations. Birmingham, Wrox Press.

[14] Freebxml.org (Accessed on 10/04/2009). ebXML Registry/Repository Overview. http://ebxmlrr.sourceforge.net/wiki/index.php/Overview

[15] OASIS/ebXML Registry Technical Committee (2002). OASIS/ebXML Registry Services Specification v2.0. http://www.oasis-open.org/committees/regrep/documents /2.0/specs/ebrs.pdf

[16] ISO TC 154 (Accessed on 10/04/2009). Processes, data elements and documents in commerce, industry and administration. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_tc_browse.htm?commid=53186&published=on

[17] OASIS Standard ebXML Business Process Technical Committee (2006). ebXML Business Process Specification Schema Technical Specification v2.0.4. http://docs.oasis-open.org/ebxml-bp/2.0.4/OS/spec/ebxmlbp-v2.0.4-Spec-os-en.pdf

[18] OASIS ebXML Joint Committee (2006). The Framework for eBusiness. http://www.oasis-open.org/committees/download.php/17817/ebxmljc-WhitePaper-wd-r02-en.pdf

[19] OASIS ebBP Technical Committee (Accessed on 11/04/2009). Frequently Asked Questions (FAQs). http://www.oasis-open.org/committees/download.php/21792/ ebxmlbp -v2.0.4-faq-os-en.htm

[20] OASIS ebXML Collaboration Protocol Profile and Agreement Technical Committee (2002). Collaboration-Protocol Profile and Agreement Specification Version 2.0. http://www.oasis-open.org/committees/download.php/204/ebcpp-2.0.pdf

[21] OASIS Universal Business Language Technical Committee (Accessed on 12/04/2009). Frequently Asked Questions (FAQs). http://www.oasis-open.org/ committees/ubl/faq.php
[22] OASIS Universal Business Language Technical Committee (2006). Universal Business Language v2.0. http://docs.oasis-open.org/ubl/os-UBL-2.0/UBL-2.0.html
[23] Supply-Chain Council (2008). SCOR 9.0 Overview. http://www.supply-chain.org/galleries/public-gallery/SCOR%209.0%20Overview%20Booklet.pdf
[24] Supply-Chain Council (Accessed on 17/04/2009). SCOR Model. http://www.supply-chain.org/cs /root/scor_tools_resources/scor_model/scor_model
[25] Lukka, K. (2003). The constructive research approach. In: L. Ojala, O-P. Hilmola. (eds.) Case study research in logistics. Publications of the Turku School of Economics and Business Administration, Series B 1: 2003, p. 83-101.
[26] Yovits, M. C. and M. V. Zelkowitz (1995). Advances in computers, volume 40. London, Academic Press.
[27] Berners-Lee, T. (1996). Universal Resource Identifiers -- Axioms of Web Architecture. http://www.w3.org/DesignIssues/Axioms.html
[28] O'Reilly, T. (2003). REST vs. SOAP at Amazon. http://www.oreillynet.com/pub /wlg/3005
[29] Google Code (Accessed on 23/04/2009). Google SOAP Search API. http://code.google.com/apis /soapsearch/
[30] W3C Candidate Recommendation (2009). SKOS Simple Knowledge Organization System Reference. http://www.w3.org/TR/skos-reference/
[31] W3C Recommendation (2004). Resource Description Framework (RDF): Concepts and Abstract Syntax. http://www.w3.org/TR/rdf-concepts/
[32] W3C Recommendation (2004). OWL Web Ontology Language Reference. http://www.w3.org/TR /owl-ref/

Appendix A – Glossary

ABIE	Aggregate Business Information Entity
ACL	Access Control List
API	Application Programming Interface
BBIE	Basic Business Information Entity
Cac	Common Aggregate Components
Cbc	Common Basic Components
CORBA	Common Object Requesting Broker Architecture
CPA	Collaboration Protocol Agreement
CPP	Collaboration Protocol Profile
CRUD	Create, Read, Update, and Delete
DCOM	Distributed Component Object Model
DFD	Data Flow Diagram
ebBP	ebXML Business Process
ebBPSS	ebXML Business Process Specification Schema
ebCPPA	ebXML Collaboration Protocol Profile and Agreement
e-Business	Electronic Business
ebXML	Electronic Business eXtensible Markup Language
EDI	Electronic Data Interchange
ERP	Enterprise Resource Planning
HTTP	Hypertext Transfer Protocol
ICT	Information and Communication Technology
ISO	International Organization for Standardization
OASIS	Organization for the Advancement of Structured Information Standards
OWL	Web Ontology Language
RDF	Resource Description Framework
REST	REpresentational State Transfer
RPC	Remote Procedure Call
SCM	Supply Chain Management
SCOR	Supply-Chain Operations Reference
SKOS	Simple Knowledge Organization System
SOAP	Simple Object Access Protocol (v1.1, discard in v1.2)
SQL	Structured Query Language
SSADM	Structured Systems Analysis and Design Method
UBL	Universal Business Language
UN/CEFACT	United Nations Centre for Trade Facilitation and Electronic Business
URI	Universal Resource Identifier
WMS	Warehouse Management System
XML	Extensible Markup Language