EXAMENSARBETET 2007
Information Technology

Using Web Services for Transparent Access to Distributed Databases

Héctor Cardenas
Jan Schneider
José Alfonso Talamantes

Detta examensarbete är utfört vid Ingenjörshögskolan i Jönköping inom ämnesområdet datateknik. Arbetet är ett led i teknologie magisterutbildningen med inriktning informationsteknik. Författarna svarar själva för framförda åsikter, slutsatser och resultat.

Handledare: Vladimir Tarassov
Examinator: Kurt Sandkuhl

Omfattning: 20 poäng (D-nivå)
Datum:
Arkiveringsnummer:
Using Web Services for Transparent Access to Distributed Databases

Héctor Cardenas
Jan Schneider
José Alfonso Talamantes

MASTER THESIS 2007
COMPUTER ENGINEERING
Abstract

This thesis consists of a strategy to integrate distributed systems with the aid of web services. The focus of this research involves three subjects, web services and distributed database systems and its application on a real-life project.

For defining the context in this thesis, we present the research methodology that provides the path where the investigation will be performed and the general concepts of the running environment and architecture of web services.

The mayor contribution for this thesis is a solution for the Chamber Trade in Sweden and VNemart in Vietnam obtaining the requirement specification according to the SPIDER project needs and our software design specification using distributed databases and web services.

As results, we present the software implementation and the way our software meets and the requirements previously defined. For future web services developments, this document provides guidance for best practices in this subject.
Acknowledgements

We would like to thank the people who shared their valuable knowledge and inspiration for the creation of this thesis. Our teachers and mentors.

Our special thanks to Vladimir Tarasov and Kurt Sandkuhl, for their representation and guidance. Maria Karlberg, representing the Chamber Trade of Sweden, for providing the needs of the chamber trade in the scope of the project.

Personal Acknowledgments

Héctor Cárdenas

I want to thank all of my friends in Mexico and most of all my friends José Talamantes and Jan Schneider. Finally I want to dedicate this thesis to my family, Benjamín, Maricela and Presysla. Thank you for everything you have done.

Jan Schneider

This work would not have been possible without the help of my teammates and friends Héctor Cardenas and José Alfonso Talamantes. I would also like to thank my family for all their love and support: Gisela, Elena, Helen and Hans.

José Talamantes Pavón

The competition of this master thesis was only possible with the support and help of my team and best friends: Hector Cardenas and Jan Schneider.

To my family that has been a constant source of motivation and love. I dedicate this thesis to them: Margarita Pavón, Carlos Talamantes, Norma Talamantes, Karla Talamantes. I am grateful for believing in all my decisions and walking always by my side.
Key words

Distributed Database, Web Service, Chamber Trade, VNemart, XML, SOA, IIS, Dot Net Framework, Information.
Contents

1 Introduction .................................................................................. 1
  1.1 BACKGROUND ........................................................................... 1
  1.2 PURPOSE/OBJECTIVES ................................................................ 1
    1.2.1 Research Questions ............................................................. 1
    1.2.2 Theoretical Purpose ............................................................ 1
    1.2.3 Practical Purpose ............................................................... 1
  1.3 TIME PLAN (APPENDIX A) ......................................................... 2
  1.4 LIMITATIONS ........................................................................... 2

2 Method ......................................................................................... 3

3 Theoretical Background ................................................................... 4
  3.1 TECHNOLOGIES FOR INTEGRATION OF HETEROGENEOUS DISTRIBUTED SYSTEM .......... 4
    3.1.1 Application Server ............................................................... 4
    3.1.2 Internet Information Services (IIS) ....................................... 4
    3.1.3 Service-oriented architecture (SOA) definition ....................... 5
    3.1.4 Technologies for Service Oriented Architecture .................... 6
    3.1.5 Distributed Database Integration ......................................... 10
    3.1.6 Integration methodologies .................................................. 11
  3.2 DATA SEARCH IN DATABASES WITH HETEROGENEOUS DATA SCHEMAS ................... 12
    3.2.1 The impact of data integration on the cost and benefit of information systems ............ 12
    3.2.2 Quality-driven Integration of Heterogeneous Information Systems ......................... 12
    3.2.3 Information Modeling for Computerized Manufacturing ................. 12
    3.2.4 Integration Of Application Systems By Modelling Information Shared Among Applications ...... 12
    3.2.5 A SERVICE-ORIENTED Integration Approach .......................... 12
    3.2.6 Healthcare Information System Integration: A Service-Oriented Approach............. 12

4 Realization ..................................................................................... 13
  4.1 RESEARCH QUESTIONS ................................................................ 13
  4.2 HYPOTHESIS DEFINITION ......................................................... 13
  4.3 METHODOLOGY VALIDATION ..................................................... 13
  4.4 SOFTWARE REQUIREMENT SPECIFICATION .................................. 13
    4.4.1 Purpose .............................................................................. 13
    4.4.2 Scope ................................................................................ 13
    4.4.3 Overall Description ........................................................... 13
    4.4.4 Specific Requirements ........................................................ 15
    4.4.5 Functional requirements ...................................................... 15
    4.4.6 Non functional requirements ............................................... 20
    4.4.7 System architecture ............................................................ 20
  4.5 SOFTWARE DESIGN SPECIFICATION ........................................... 21
    4.5.1 High level system architecture ............................................. 21
    4.5.2 Design Description Information Content .................................. 21
    4.5.3 Web Service ....................................................................... 22
    4.5.4 Competence Structure ......................................................... 23
    4.5.5 Interface and Functionality .................................................. 24
    4.5.6 Functionality Design ........................................................... 26
    4.5.7 Search Companies ............................................................... 26
    4.5.8 Return Companies .............................................................. 28
    4.5.9 Translator ........................................................................... 28
    4.5.10 Upload Competence .......................................................... 29
    4.5.11 Validate Competence ......................................................... 29
5 Results .......................................................................................................................................... 31
  5.1 IMPLEMENTATION .................................................................................................................. 31
      5.1.1 Web Service ...................................................................................................................... 31
      5.1.2 Database Integration ........................................................................................................ 37
  5.2 HYPOTHESIS TESTING ....................................................................................................... 39
  5.3 BEST PRACTICES .................................................................................................................... 46
      5.3.1 Installation ........................................................................................................................ 46
      5.3.2 Solution to Installation ...................................................................................................... 47
      5.3.3 Webconfig ........................................................................................................................ 47
      5.3.4 DNS .................................................................................................................................. 48
      5.3.5 Compilation problem ......................................................................................................... 48
      5.3.6 Firewall ................................................................................................................................ 49

6 Conclusion .................................................................................................................................... 50

7 References .................................................................................................................................... 51

8 Appendix ...................................................................................................................................... a
  8.1 TIME PLAN ................................................................................................................................. A
List of Figures

FIGURE 1.2.2-1 (APPLICATION SERVER OVERVIEW) ........................................ 4

FIGURE 3.3.1-1 (EVOLUTION OF SOA)[7] .............................................................. 6

FIGURE 3.4.1-1 (STRUCTURE OF CORBA OBJECT REQUEST INTERFACES) .............................................................................................................. 7

FIGURE 3.4.2-1 (COM ARCHITECTURE) .................................................................. 7

FIGURE 3.4.3-1 (DCOM ARCHITECTURAL MODEL) ............................................. 8

FIGURE 3.4.4-1 (WEB SERVICE ARCHITECTURE) ................................................. 9

FIGURE 4.3.3-1 (CHAMBER TRADE WS INTERACTION) ....................................... 14

FIGURE 4.3.3-2 (COMPONENTS INTERACTION IN COMPETENCE DATABASE) .............................................................................................................. 14

FIGURE 4.3.5-1 (USE CASE CHAMBER TRADE WEB SERVICE) ................. 16

FIGURE 4.3.5-2 (USE CASE CHAMBER TRADE WEB APPLICATION)............. 16

FIGURE 4.4.3-1 (CHAMBER TRADE WEB SERVICE COMPONENT) .......... 22

FIGURE 4.4.4-1 (CHAMBER TRADE SYSTEM COMPONENTS) .......................... 23

FIGURE 4.4.4-2 (GENERAL SYSTEM ARCHITECTURE DIAGRAM) ............. 23
FIGURE 4.4.6-1 (SEQUENCE DIAGRAM SEARCH COMPANIES IN CHAMBER TRADE)..................................................................................................................26

FIGURE 4.4.6-2 (STATE CHART DIAGRAM SEARCH COMPANIES IN CHAMBER TRADE)..................................................................................................................26

FIGURE 4.4.7-1 (SEQUENCE DIAGRAM SEARCH COMPANIES IN CHAMBER TRADE & VNEMART).................................................................................................27

FIGURE 4.4.7-2 (SEQUENCE DIAGRAM SEARCH COMPANIES IN CHAMBER TRADE & VNEMART).................................................................................................27

FIGURE 4.4.8-1 FIGURE 4.4.8-2 (SEQUENCE DIAGRAM RETURN COMPANIES IN CHAMBER TRADE & VNEMART).................................................................28

FIGURE 4.4.8-3 (STATE DIAGRAM RETURN COMPANIES IN CHAMBER TRADE & VNEMART)...........................................................................................................28

FIGURE 4.4.9-1 (STATE MACHINE TRANSLATOR DIAGRAM) ....................................................28

FIGURE 4.4.10-1 (SEQUENCE DIAGRAM UPLOAD COMPETENCE) .................29

FIGURE 4.4.11-1 (SEQUENCE DIAGRAM VALIDATE COMPETENCE) ..........29

FIGURE 4.4.11-2 (SEQUENCE DIAGRAM VALIDATES COMPANIES CHAMBER TRADE & VNEMART).................................................................................................30

FIGURE 5.1.1-1(SERVICE.ASMX)..................................................................................31

FIGURE 5.1.1-2(SERVICE.ASMX)..................................................................................31

FIGURE 5.1.1-3(WEB.CONFIG)..................................................................................31
FIGURE 5.1.1-4 (SERVICE.CS) .......................................................... 32

FIGURE 5.1.1-5 (SERVICE.CS) .......................................................... 32

FIGURE 5.1.1-6 (SERVICE.CS) .......................................................... 32

FIGURE 5.1.1-7 (SERVICE.CS) .......................................................... 32

FIGURE 5.1.1-8 (SERVICE.ASMX) .................................................... 32

FIGURE 5.1.1-9 (SERVICE.CS) .......................................................... 33

FIGURE 5.1.1-10 (XML SEARCH) .................................................... 35

FIGURE 5.1.1-11 (XML RESULT) ...................................................... 36

FIGURE 5.1.2-1 DATA STRUCTURE OF CLASSIFICATION SCHEMA .... 38

FIGURE 5.1.2-1 (DATA INTERCEPTION BY THE WEB SERVICE) ........ 39

FIGURE 5.1.2-2 (RESULTS RETRIEVED BY THE WEB SERVICE) ....... 40

FIGURE 5.1.2-3 (DATA IDENTIFICATION AND QUERY CREATION) .... 41

FIGURE 5.1.2-4 (CREATION OF THE QUERY) ................................. 43

FIGURE 5.1.2-5 (DEMONSTRATION OF THE TRANSLATION) ........... 43

FIGURE 5.1.2-6 (XML DOCUMENT BY VNEMART) .......................... 44
FIGURE 5.1.2-7 (RESULTS DISPLAYED FROM THE REQUESTED QUERY) .................................................................44

FIGURE 5.1.2-8 (RESULTS BY THE WEB SERVICE) .........................................................................................44

FIGURE 5.1.2-9 (SAMPLE CODE OF THE WS INTEGRATION) .................................................................45

FIGURE 5.1.2-10 (RESULTS RETRIEVED BY THE WEB SERVICE) .........................................................46

FIGURE 5.3.3-1 (WEB.CONFIG ERROR) .......................................................................................................47

FIGURE 5.3.3-2 (CORRECT SETTINGS OF WEB.CONFIG) ...........................................................................48

FIGURE 5.3.5-1 (COMPILATION ERROR) ...................................................................................................49
List of Abbreviations

- **FIFO**.- First In First Out.
- **ICT**.- Information and communication technologies.
- **JEE**.- Java Enterprise Edition.
- **LAN**.- Local Area Network.
- **SOA**.- Service Oriented Architecture.
- **SOAP**.- Simple Object Access Protocol.
- **WAN**.- Wide Area Network.
1 Introduction

1.1 Background

Nowadays many distributed systems are inherently heterogeneous and geographically dispersed. Such systems emerge as a result from the cooperation between organizations inside the country or between different countries. Bridging information systems for business support from different organizations imposes a challenge because of diverse underlining technologies and data structures used. To solve this problem, Service Oriented Architecture implementations such as Web Services are often employed for being an efficient approach to hide heterogeneity and provide seamless access to complex information systems.

The final thesis work is linked to the Chamber Trade project group Chamber Trade Business to Business AB and School of Engineering of Jönköping University within the SPIDER project. Before the realization of this project, the databases and application form VNmart and Chamber Trade were not connected. Presently, there is no interconnection between the two web applications and a search is only made in Chamber Trade or in VNemart respectively.

The main intention of this project is to support establishment of business relationships between enterprises in Sweden and Vietnam by providing ICT tools for finding appropriate partners.

1.2 Purpose/Objectives

1.2.1 Research Questions

• How can heterogeneous databases aided by a web based interface be integrated to provide uniform access for the users?
• How can a SOA, service oriented architecture, be used to request information from two different databases?
• Is there a method to uniformly search for data in databases with different schemas?

1.2.2 Theoretical Purpose

Since global enterprises are emerging and strategically partnerships are being created; data gets stored in different databases located all over the world. Gaining access to the data in a location free environment brings value to the organizations and an advantage to their users.

Based on the previous statement methods for integrating and consulting different databases using a web application will be analyzed and discussed. For accomplishing the information request from a database in service oriented architecture, in an efficient way, an investigation for the best method will be performed.

1.2.3 Practical Purpose

The purpose of the project work is to develop a web service for connecting the databases used in the SPIDER project. The web service will establish a transparent connection between the Chamber Trade database in Sweden and VNemart database in Vietnam, hosted by Vietnamese Chamber of Commerce and Industry. Both databases store information from enterprises willing to find business partners and allow users to register their business or search for companies. Upon completion of the project, the user will retrieve information from both databases simultaneously with one search. The databases describe products/services produced
by companies using industrial classifications. The mapping between the classifications will be provided. The database and web service to be developed shall contain the following major components:

- A search component making it possible for the VNemart web application to remotely retrieve data from the Chamber Trade enterprise and competence databases, either simultaneously or independently.
- A translation component mapping the Chamber Trade industrial classification onto the VNemart classification before sending retrieved data back.
- Although not formally a part of the web service, a web service consumer component is to be added to the Chamber Trade web application to make a remote search in VNemart when a search in Chamber Trade is initiated and to integrate data remotely retrieved from VNemart with data locally retrieved from Chamber Trade before presenting them to the Chamber Trade user.

1.3 Time Plan (Appendix A).

1.4 Limitations

This thesis will only cover the connections between the databases using web services. A Web Service will be created for the Chamber Trade in Sweden; will not include the database upload. A test web service will be created for replacing the web service needed to be created by the VNemart for validation and for testing.
2 Method

In order to obtain Master of Science, with a major in Computer Engineering, specializing in Information Technology, the master thesis involves a real-life solution and includes a theoretical background for its development. A situation concerning connections between Sweden and Vietnam was presented for our development. The methodology for presenting a solution was delimited by different factors. The main factor brought into consideration was the client requirement analysis and the technology presently in use by the Chamber Trade. Literature studies and Software Engineering practices were used to the development of the project.

To solve the problem and the situation presented by the Chamber Trade, we followed the next steps:

- **Problem Identification.** - A situation was stated by the chamber trade and we identify the goals they want to achieve.
- **Client requirement analysis.** - The chamber trade presented their perspective and specific necessities were identified.
- **Establishment of limitations and constraints of the project.** - Every project has a specific beginning and a specific end, as well as the tools, procedures and rule for its creation. In this case, were proposed by the chamber trade in collaboration with the School of Engineering and our team.
- **Hypothesis creation.** - A statement was created about the relationship between two or more variables caring implications for testing. For this scenario, our variables were the creation of a Web Service and databases and the interaction between them.
- **Development.**
  - **Software Requirement Specification.** - A tailoring of the project into the IEEE Standard
  - **Design Specification.** - Software engineering tools were followed, such as UML, to bring a better understanding of the solution.
  - **Literature review for developing solution.** - In order to support our previous skills and experience, literature was reviewed and used to generate the needed knowledge for developing the chamber trade request.
  - **Programming.** - Algorithms and knowledge were coded into create files that can be computed.
  - **Testing.** - The computed results were compared to the theoretical expectations
- **Conclusions.** - Summary of the interpretation of results, expectations and experiences of the complete process in order to achieve its objectives.
3 Theoretical Background

3.1 Technologies for integration of heterogeneous distributed system.

3.1.1 Application Server

Application server is the definition given to a server computer or a computer network dedicated to software applications, especially web based applications. An application server provides an efficient development platform, improving runtime efficiency by simultaneously managing access to distributed resources for several web based applications keeping the applications consistent and scalable for an increasing number of users. By isolating the operating system, the development of the web application, relates to the business logic and data access rather than technical conditions such as resource allocation or memory management. [1][2]

Application Servers are typically implemented on a multi-tiered architecture where business applications are separated into components that can be running on different machines. The An Server provides the framework for a client to connect to a backend source, execute the applications logic, and return the result to the client. The location of the application server is the middle-tier in the three-tier (three-layer) in the computing architecture model; this architecture is a physical separation between the presentation tier, the application tier and the data tier.[1][2][4]

![Application Server Overview](image)

Figure 3.1.1-1(Application Server Overview)

3.1.2 Internet Information Services (IIS)

Microsoft Internet Information Services is a set of Internet services (FTP, SMTP, NNTP and HTTP/HTTPS) for servers using Microsoft Windows, it is the second most popular web server used over the internet behind Apache HTTP Server.
3.1.2.1 IIS 6 Main features

3.1.2.2 IIS Request Processing Architecture

- **HTTP Listener process.** -The main process here is the driver called Http.sys and is used for parsing and catching. It listens for requests and sends them to the worker processes.

- **Worker processes.** -This permits the server to run different applications in the same pool, they become active when demanded and stop using resources when they are inactive.

- **Application Pools.** -This is a group of web applications; each pool is serviced by one or more worker process.

- **Application pool request queue.** - After the request is taken by the Http.sys, the request must wait in a FIFO queue and then the workers attend the request from the queue. Each Application pool has one application pool queue.

3.1.2.3 IIS Security Architecture

- **Capability lockdown.** - In the IIS version 6 all the capabilities must be specifically enabled, it only serves

- **Privilege changes.** - IIS 6 uses the account Network Service which has low privileges by default.

- **Tool and file restrictions.** - IIS won’t run request coming from command= line and it will not service request coming from invalid files.

- **Authentications enhancements.** - between the authentications types IIS 6 has the most important one is the integration of .NET passport identification. [5]

3.1.3 Service-oriented architecture (SOA) definition

Service Oriented Architecture is a service-oriented architecture style witch utilizes technologies, which provide a collection of reliable services to connect and communicate software applications even over different platforms. The communication can involve either simple data passing or it could involve two or more services coordinating some activity. The usage of SOA brings the possibility to develop richer and more advanced applications for information systems.

One of the first service-oriented architecture was with the use DCOM or Object Request Brokers based on CORBA specification. After the creation of the Web Services, new developments according to SOA architecture brought the possibility of communicating enterprises according to the available service.

Related to service-oriented architecture the term service is known by a remote function that is self-contained, well-defined and does not depend on the context or state of other services. More specific the logical grouping of components to satisfy a business is called service. A service has a behavior that can be implemented and provided by a component for use by another component. The advantage of SOA is that operating systems, network protocols and programming languages that are used between the components are not part from the developers concern, bringing the possibility to create reusable software components.[6]

3.1.3.1 Evolution of SOA

The history of SOA can be traced to the monolithic application era, at that moment large scale applications were deployed. The as the next figure indicates compares the evolution form unstructured code to object oriented as the foundation of the Service oriented Architecture.
3.1.4 Technologies for Service Oriented Architecture

Since Service Oriented Architecture (SOA) is an evolution of Object Oriented (component based architecture for separating functionality into individual objects that work together) and Distributed Systems such as J2EE, CORBA, and DCOM. Defining Web Services, .NET, J2EE and, CORBA as specialized SOA implementations that represent the core aspects of a service-oriented approach to SO architecture.

3.1.4.1 CORBA: Common Request Broker Architecture

Common Request Broker Architecture is not so much a distributed system but provides a framework where objects can communicate with platform and language freedom in a distributed manner. The collection of objects that isolate the requestors of services by a well defined encapsulating interface is commonly known as an object system, in this particular situation, clients are isolated from the implementation of the service, data representation and coding.

CORBA specifications have been stated up by the Object Management Group who defines the Interface definition language, the core API that includes the classification of a communication infrastructure based on Object Request Broker for distributed applications, and an internet Inter-ORB protocol that is the TCP/IP based communication protocol.[8]
3.1.4.2 COM: Component Object Model

Component Object Model as a family of Microsoft family Operating Systems is used by developers to create reusable, link components in other to develop reusable software, link components together, and to build applications taking advantage of the Windows services. COM Family COM+, Distributed COM (DCOM) and ActiveX Controls. [9]

COM is a component-based software architecture which allows components from variety of applications to be combined and built into a new higher-level software application. COM defines a standard for component interoperability. It is independent from particular programming language, available on multiple platforms, and extensible. [9]
3.1.4.3 DCOM: Distributed Component Object Model

Distributed Component Object Model which originally was called “Network OLE” extends COM so as to support communication among objects among network boundaries (LAN, WAN, and the Internet). It allows the components to be used at distributed environment by handling the low-level details of network protocols.

![DCOM Architectural Model](image)

Figure 3.1.4-3 (DCOM Architectural Model)

3.1.4.4 Web Services

Vendor definitions of a Web Service

**W3C.** - [A Web service is a software system 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 HTTP with an XML serialization in conjunction with other Web-related standards]. [11]

**IBM.** -[Web Services are self-contained, modular applications that can be described, published, located, and invoked over a network, generally, the World Wide Web.] [12]

**Sun.** - [A Web service is a modular piece of code on the Internet that provides one or more business functions, and that can be discovered and used on demand.] [13]

**Microsoft.** - [A Web Service is programmable application logic accessible using standard Internet protocols. Web services combine the best aspects of component-based development and the Web. Like components, Web services represent black-box functionality that can be reused without worrying about how the service is implemented. Unlike current component technologies, Web services are not accessed via object-model-specific protocols, such as the distributed Component Object Model (DCOM), Remote Method Invocation (RMI), or Internet Inter-ORB Protocol (IIOP). Instead, Web services are accessed via ubiquitous Web protocols and data formats, such as Hypertext Transfer Protocol (HTTP) and Extensible Markup Language (XML). Furthermore, a Web Service interface is defined strictly in terms of the messages the Web Service accepts and generates. Consumers of the Web Service can be implemented on any platform in any programming language, as long as they can create and consume the messages defined for the Web Service interface.]
3.1.4.4.1 Architecture
Web Services are distributed software components that can be accessed through standard web protocols. The advantage of Web Services is that they can be consumed by any application that is able to parse and XML-formatted stream transmitted through http. XML is the key technology in web services.

To parse an XML-formatted stream transmitted through HTTP channels, XML is the key technology used in Web Services and is used in the following areas of the Microsoft .NET Web Services framework.

A web service is a component running on a web server that communicates to the world with standard Internet protocols such as: HTTP GET, HTTP POST, and SOAP (Simple Object Access Protocol).

![Figure 3.1.4-4 (Web Service Architecture)]

3.1.4.4.2 HTTP GET and HTTP POST
HTTP GET and HTTP POST use HTTP as their underlying protocol. These both methods encode request parameters as name/value pairs in the HTTP request. The GET method creates a query string and appends it to the script's URL on the server in order to handle the request. For the POST method, the name/value pairs are passed in the body of the HTTP request message.

3.1.4.4.3 SOAP
SOAP is a mechanism for passing messages between the clients and servers. Clients are web services consumers, and the servers are the web services. The clients send an XML-formatted request message to the server to make use of the service. The server responds by sending back another XML-formatted message. In the SOAP specification it is described the format of these XML requests and responses.

3.1.4.4.4 Web Services Description Language (WSDL)
Web service clients need to have a description of the method calls, or the interface that the web service supports, in order to interact with it. This is defined in a document that has an
XML schema called WSDL (Web Services Description Language). WSDL files describe an interface's method calls and the list of in and out parameters for the particular call. [15]

3.1.4.5 JEE Web Services

JEE by java, uses the JAX-WS technology to build the web services. JAX-WS stands for Java API for XML web services. JAX-WS provides the functionality to write message-oriented as well as RCP-oriented web services.

The invocations in JAX-WS are represented by SOAP. The SOAP specification defines the conventional rules for the invocations and responses of web services.

SOAP messages are complex but JAX-WS hides the complexity to the developer.

On the server side, the developer defines the web service methods and operations using an interface written in Java programming language.

The clients of the web service create a local object representing the service and then the client invokes the methods. With JAX-WS the developer has not to worry about parsing or generating the SOAP files, the JAX-WS runtime system is the one doing that.

In JAX-WS are 9 basic steps to create a web service and use it:

1. Code the implementation class.
2. Compile the implementation class.
3. Use “wsgen” to generate the artifacts required to deploy the service.
4. Package the files into a WAR file.
5. Deploy the WAR file. The web service artifacts (which are used to communicate with clients) are generated by the Application Server during deployment.
6. Code the client class.
7. Use “wsimport” to generate and compile the web service artifacts needed to connect to the service.
8. Compile the client class.
9. Run the client.[16]

3.1.4.6 .Net Framework Web Services

All Web Services in .Net have the following elements:

1. A asmx file for the web service. This must contain the <% webservice ... %>
   directive, as well as the class that provides the web service implementation. To the Web Service clients, this asmx file is the entry point to the Web Service. Instead of deploying as it is with JEE with .NET framework this file should be in a virtual directory with permits.

2. Inherit from the Web Service class of the System.Web.Services namespace. This allows the derived class to access all the normal ASP objects exposed in the Web Service base class.

3. Tag the methods that are going to be accessed on the web with WebMethod attributes.

4. A configuration file called web.config is placed in the same directory as the asmx file. This configuration file controls many settings from the virtual directory. [17]

3.1.5 Distributed Database Integration

A new problem that has appeared in the context of distributed databases has to do with the integration between them. Distributed databases are collections of data that is used in one
Theoretical Background

system, but this data is spread in different computers that are in the same network. They are classified in two mayor categories: Homogeneous and heterogeneous. Homogeneous databases share the same structure and schema. Heterogeneous databases have different schemas and structures. [18]

Nowadays database integration methodologies do not really deal with heterogeneity. They assume that before integration starts all schemas are translated into equivalent schemas based on a unique data model. [19]

The integration of heterogeneous schemas is a two phase process. In the first step, all commonalities and discrepancies between the schemas are determined. This is also called investigation phase and it is done manually. The Database administrator examines the schemas and defines a set of inter-schema correspondences. In the second phase the integrated schema is built semi-automatically. The database administrator needs to solve conflicts among input schemas whenever the integrator does not know how to do it. Conflicts arise every time corresponding concepts are modeled with different representations. [19]

3.1.6 Integration methodologies

3.1.6.1 Manual integration methodologies.

The first database integration methodology was developed by Motro and Buneman(1981) and it was manual. Here the integrated schema, called superview, is built over existing schemas of local databases. The input schemas and the superview are described using a functional model improved with generalization. The superview is the result that comes after the database administrator is involved in the process of editing a schema. This process defines a sequence of operations; each one performs a modification or a restructuring transformation that is applied to the initial schema. [19]

3.1.6.2 Semi-automatic integration methodologies.

Research has been done in the area of automating the database integration. These methodologies are proposed for both view and database integration. They use assertions in order to state correspondences between objects among different schemas. Each type of assertion has an integration rule, so the system knows what to do to build the integrated schema from the initial ones. The database administrator interacts only when irresolvable conflicts are detected and shows the system how to solve the conflict. [19]

Model Independent Assertions for Integration of Heterogeneous Schemas

It starts with the hypothesis that commonalities between input schemas have been identified and checked by the database administrator and the users. These commonalities are defined using inter-schema correspondence assertions. An inter-schema correspondence assertion is a declarative statement asserting that something in one schema is related to something in other. Assertions precisely identify which semantic, descriptive, and structural conflicts exist within each correspondence.

The integrator system receives the input from two or more schemas and the assertions between them. The set of assertions is scanned and ordered for processing. Each assertion is then considered and the appropriate integration rule is applied, taking in consideration the data models of the input schemas. The Integration rules define which constructs have to be
built into the integrated schema and how these constructs are mapped to the corresponding constructs in the input schemas. [19]

3.2 Data search in databases with heterogeneous data schemas.

3.2.1 The impact of data integration on the cost and benefit of information systems

Information Systems and Database integration is not an easy task, as Dale L. Goodhue, Michael D. Wybo, Lauri J. Kirsh discuss, sometimes it is easier to do a new big integrated system than integrating all the parts. A Cost Benefit analysis is suggested before starting the integration. [20] One of the most important part of the integration is the selection of good data sources. Once the selection of data is done, different several approaches can be used to do this task. [21]

3.2.2 Quality-driven Integration of Heterogeneous Information Systems

Felix Naumann, Ulf Láser, Johann Christoph Freytag suggest to create manually the correct queries of the different databases and then they integrate the data. [21]

3.2.3 Information Modeling for Computerized Manufacturing

For Cheng Hsu and Laurie Ratnner A way to solve the problem of integrating different databases is to create another database called meta database which describes the structure and information model of the databases that need to be integrated. [21]

3.2.4 Integration Of Application Systems By Modelling Information Shared Among Applications

Parames Ghosh and George Feuerlicht tell that distributed objects can be used for integration between heterogeneous Systems and a strategy used for is to create systems with an architecture that considers only the sharable features of the individual application systems. So an object can be developed on top of each application database to export only these sharable features. With this approach one can achieve that objects form a homogeneous set, participate in an integrating application system. [21]

3.2.5 A SERVICE-ORIENTED Integration Approach

According to Samuel K. Moore, to integrate data between different databases a software that understands the format of associated systems will be able to translate queries that are used by the individual databases. [21]

3.2.6 Healthcare Information System Integration: A S. Oriented Approach.

Weiping Wang, Mingming Wang and Shijun Zhu used a Service-Oriented Integration Approach, the main idea of this method is to introduce a new component to distributed systems bringing service as a management role. This component provides a uniform interface of service management, and the manufactures made programming interface into service component. [25].
4 Realization

The final thesis work will focus on approaches and methods for implementing web services and using them for integration of physically dispersed databases with different data structures.

4.1 Research Questions

- How can heterogeneous databases aided by a web based interface be integrated to provide uniform access for the users?
- How can a SOA, service oriented architecture, be used to request information from two different databases?
- Is there a method to uniformly search for data in databases with different schemas?

4.2 Hypothesis Definition

Web services can be used to create a component for intercepting; identifying, integrating and showing data from different databases while being requested throw a web application as a transparent procedure for the user.

4.3 Methodology Validation

After the research process; SOA, in specific, web services, are the newer and most powerful architecture for integrating distributed information systems. Besides this technology was adapted and customized to client specific requirements.

An important part of the development was the integration of databases situated in different locations and based on different schemas. None of the reviewed integrations were suitable for this scenario forcing us to base the integration in our own methodology.

4.4 Software Requirement Specification

4.4.1 Purpose

The main intention of the project is to support the business link and relationships between enterprises in Sweden and Vietnam by providing a Web service that allows integrating and presenting information between the Chamber Trade’s database in Sweden and the VNEmart database in Vietnam. The project will retrieve and present information about registered persons that can establish the business relationships.

4.4.2 Scope

The system we are going to develop will be a bridge between the Chamber Trade and VNEmart web applications. It will also work to let individuals store their competence information. And the administrator will be able to validate that information.

4.4.3 Overall Description

4.4.3.1 Product perspective
The Chamber Trade (Sweden) owns a web application running where it is possible to consult data from the different registered enterprises.

The VNemart (Vietnam) should have a Web service running and should be able to provide data from the different enterprises registered in its database.

For this project to be successful it is important to make modifications in the chamber trade web application. And have a common interface with the VNemart Web service.

In figure 3-1 there is showed how the Chamber Trade Web service is going to interact with the other existing components such as: Chamber Trade Web application, VNemart Web Service and Chamber Trade Database.

In figure 3-2 it is explained the different components that will work with the competence database.
4.4.3.2 **Product Users Characteristics**

- Enterprises using the Chamber Trade Database in Sweden
- VNemart Web service
- Administrator of the Chamber Trade

4.4.3.3 **Assumptions and dependencies**

- The users will access the search component through the Chamber Trade web application with the use of a web browser.
- The components of this project are going to be installed in the Chamber Trade Server. *(Running over MS Windows 2000 Server)*

4.4.4 **Specific Requirements**

4.4.4.1 **User requirements**

- Include target countries in search
- Search should be performed in all available databases
- Data coming from each particular DB should be differentiated (marked with color or a special symbol.)
- The competence DB should have the same user interface as the enterprise DB to a maximal possible extent.
- Search for supporting business competence should be separate one, simultaneous search for enterprises and supporting competence is to be avoided.
- Entries in the competence DB must be checked before they are published. In the current version, new entries are first entered in an administrative DB. After that they are validated by Chamber Trade and moved to the public DB.
- To validate a record, Chamber Trade needs a reference person or organization (e.g. a bank, migrant society, etc.)

4.4.5 **Functional requirements**

**Use Cases**
Realization

Figure 4.4.5-1 (Use Case Chamber Trade Web Service)

Figure 4.4.5-2 (Use Case Chamber Trade Web Application)
NAME: Search Companies

DESCRIPTION:
This is when a Chamber Trade user makes a search for enterprises through the Chamber Trade webpage. First the Web Service is going to make a search in the Chamber Trade Database, then ask for a request to the VNemart Web Service, and finally will returned the information to the user.

ACTORS:
Primary Actor: Chamber Trade user
Secondary Actors: VNemart Web Service, Chamber Trade Web Service, Chamber Trade Database

PRECONDITIONS:
Primary Actor enters the Chamber Trade webpage.
Chamber Trade Web Service must know the interface search interface from the VNemart Web Service in order to ask for the right information.
Chamber Trade Web Service must be running.
VNemart Web Service must be running.
Chamber Trade Database server must be running.

SUCCESS SCENARIO:
1) Primary Actor selects parameters for the search.
2) Chamber Trade Web Service intercepts parameters.
3) Chamber Trade Web Service asks information to chamber trade database according to the parameters.
4) Chamber Trade Web Service translates categories and sends a request to VNemart Web Service.
5) Chamber Trade Web Service returns requested information to Primary Actor.

EXTENTIONS:

SPECIAL REQUIREMENTS:

FREQUENCY OF OCCURRENCE:
Very Often

NAME: Search Companies in Chamber Trade

DESCRIPTION:
This is when another Web Service, in this case VNemart Web Service wants to access information about Chamber Trade.

ACTORS:
Primary Actor: VNemart Web Service.
Secondary Actors: Chamber Trade Web Service, Chamber Trade Database.

PRECONDITIONS:
Primary Actor knows the interface of the Chamber Trade Web Service.
Chamber Trade Web Service must be running.
VNemart Web Service must be running.
Chamber Trade Database server must be running.

SUCCESS SCENARIO:
1) Primary Actor sends parameters to Chamber Trade Web Service.
2) Chamber Trade Web Service look for information in Chamber Trade database.
3) Chamber Trade Web Service returns information to VNemart Web Service.

EXTENTIONS:

SPECIAL REQUIREMENTS:

FREQUENCY OF OCCURRENCE:
Very Often

NAME: Return Companies
DESCRIPTION:
When the user already found the enterprises he is interested in, the Web Service shows him the whole information about them.

ACTORS:
Primary Actor: Chamber Trade user.
Secondary Actors: VNemart Web Service, Chamber Trade Web Service

PRECONDITIONS:
Primary Actor already made a search for companies.
Chamber Trade Web Service must be running.
Chamber Trade Web Service knows the interface of the VNemart Web Service.
VNemart Web Service must be running.
Chamber Trade Database server must be running.

SUCCESS SCENARIO:
1) Primary Actor selects which companies he is interested in.
2) Chamber Trade Web Service intercepts list of companies.
3) If Companies belong to Chamber Trade Database.
   3.1) Chamber Trade Web Service ask information to Chamber Trade Database about the companies in the list.
4) If Companies belong to VNemart Database.
   4.1) Chamber Trade sends petition of the specific company to VNemart Web Service.
5) Chamber Trade Web Service return requested information to Primary Actor.

EXTENSIONS:

SPECIAL REQUIREMENTS:

FREQUENCY OF OCCURRENCE:
Very Often

NAME: Uploading Competence
DESCRIPTION:
The user registers his competence in the chamber trade Web-Application.

ACTORS:
Primary Actor: Chamber Trade user.
Secondary Actors: Chamber Trade Web-Application, Chamber Trade Database

PRECONDITIONS:
Primary Actor enters to the Chamber Trade Web-Application.
Chamber Trade Database server must be running.

SUCCESS SCENARIO:
1) Primary Actor fills a Form.
2) Primary Actor submits the Form.
3) The information is uploaded in the Chamber Trade Database as a possible Competence.
4) A page will appear to the primary Actor telling that the information was uploaded, and that the personal of the Chamber Trade will send the user a confirmation mail when his information will be published.

EXTENSIONS:
The insertion of the records in the database could not be made.
4b) A page will appear to the primary Actor telling that there was a problem and that the user should try it again later.

SPECIAL REQUIREMENTS:

FREQUENCY OF OCCURRENCE:
Often
NAME: Validating Competence
DESCRIPTION:
An administrator of the Chamber Trade will look the new candidates for competence and will validate the information.

ACTORS:
Primary Actor: Chamber Trade Administrator.
Secondary Actors: Chamber Trade Web-Application, Chamber Trade Database

PRECONDITIONS:
Primary Actor logs in to the Chamber Trade Web-Application.
Chamber Trade Database server must be running.

SUCCESS SCENARIO:
1) Chamber Trade Web-Application shows the primary actor the new candidates for competence.
2) Primary Actor selects one.
3) Primary Actor validates it.
4) If the information is valid
   4.1) The information would be uploaded to the Chamber Trade Database
   4.2) A mail will be sent to the user telling that his profile is published now.
5) If the information is not valid.
   5.1) The information will be deleted from the Chamber Trade Database.

EXTENSIONS:
The insertion of the records in the database could not be made.
4b) A page will appear to the primary Actor telling that there was a problem and that he(she) should try it again later.

SPECIAL REQUIREMENTS:
FREQUENCY OF OCCURRENCE:
Often

NAME: Searching Competence
DESCRIPTION:
The user search for competence in the database.

ACTORS:
Primary Actor: Chamber Trade user.
Secondary Actors: Chamber Trade Web-Application, Chamber Trade Database

PRECONDITIONS:
Primary Actor enters to the Chamber Trade Web-Application.
Chamber Trade Database server must be running.

SUCCESS SCENARIO:
1) Primary Actor fills a Form.
2) Primary Actor submits the Form.
3) The right information about the competence is showed.

EXTENSIONS:
SPECIAL REQUIREMENTS:
FREQUENCY OF OCCURRENCE:
Often

NAME: show specific competence
DESCRIPTION:
The user selects the specific competence that wants to look at, and the application shows it.

ACTORS:
Primary Actor: Chamber Trade user.
Secondary Actors: Chamber Trade Web-Application, Chamber Trade Database
PRECONDITIONS:
Primary Actor enters to the Chamber Trade Web-Application.
Chamber Trade Database server must be running.

SUCCESS SCENARIO:
1) Primary Actor selects some competences.
2) The right information about the competence is showed.

EXTENSIONS:

SPECIAL REQUIREMENTS:

FREQUENCY OF OCCURRENCE:
Often

4.4.6 Non functional requirements

Software Constraints
- **Description:** The technology to be used is .NET Framework.
- **Foundation:** The components will run in the Chamber Trade Web Server and it is MS Windows 2000 Server
- **Fit criterion:** the product should run in a MS Windows 2000 Server
- **Description:** The Competence Database should be implemented in MS SQL Server.
- **Foundation:** The client uses MS SQL 2000 Server.

4.4.7 System architecture

The project consists of 5 principal components:

1. The competence database created and located on the Chamber Trade server.

2. A user interface component to update and search the competence database locally (by users in Sweden), this component should be integrated with the existing user interface of the Chamber Trade enterprise database to a possible extent.

3. A search component making it possible for the VNemart web application to remotely retrieve data from the Chamber Trade enterprise and competence databases, either simultaneously or independently.

4. A translation component mapping the Chamber Trade industrial classification onto the VNemart classification before sending retrieved data back.

5. A Web service consumer component will be added to the Chamber Trade web application to integrate data retrieved from Chamber Trade database simultaneously with the VNemart database and presenting the result to the Chamber Trade user.
4.5 Software Design Specification

4.5.1 High level system architecture

4.5.1.1 Components
The architecture consists of 5 principal components:

- The competence database created and located on the Chamber Trade server.

- A user interface component to update and search the competence database locally (by users in Sweden), this component should be integrated with the existing user interface of the Chamber Trade enterprise database to a possible extent.

- A search component making it possible for the VNemart web application to remotely retrieve data from the Chamber Trade enterprise and competence databases, either simultaneously or independently.

- A translation component mapping the Chamber Trade industrial classification onto the VNemart classification before sending retrieved data back.

- A Web service consumer component will be added to the Chamber Trade web application to integrate data retrieved from Chamber Trade database simultaneously with the VNemart database and presenting the result to the Chamber Trade user.

4.5.2 Design Description Information Content
In order to have a general overview of the project the reader must look at the next diagram. This diagram shows the tree parts related to the project in which only the VNEmart Web Service is located in Vietnam and the rest of the system is hosted in Sweden. The Software developed in this project is the Chamber Trade Web service.
4.5.3 Web Service

- **LoadComponent**: This Component will load all the attributes to access the Database and the IP and interfaces to the other Web Services.
- **TranslatorComponent**: This Component will translate the Chamber Trade classification to the VNemart Classification.
- **QueryCreatorComponent**: This component will create the queries according to the parameters received by the users.
- **StringCreatorComponent**: This component creates the string that will be sent to the web application that called the web service with all the information needed.
- **CreateXMLComponent**: This component creates an XML document that will be sent to the VNemart web service.

Figure 4.5.3-1 (Chamber Trade Web Service Component)
4.5.4 Competence Structure

- **UploadCompetence.** - This component will get the information given by the user and save it as new competence in the database.
- **ValidateCompetence.** - This component will let the administrator to check the new competences in order to validate or erase them.
- **SearchComponent.** - This component allowed the user to search for competence.

In order to know more details about the structure and design of the Chamber Trade Web Service, The reader can use the next diagram for a better understanding of the projects development.
4.5.5 Interface and Functionality

4.5.5.1 External Interfaces

Name: CTsearch
Description of purpose: The web application of the Chamber Trade will call the Web service through this interface in order to get the results of the search according to the different parameters selected by the user.
Source of Input: Chamber Trade web application.
Input parameters:
- LookingFor
- FreeWord
- Country
- Area
- Employees
- Classification

Name: VNsearch
Description of purpose: The Web service of the VNemart will call the Web service using this interface in order to get the results of the search according to the different parameters selected by the user.
Source of Input: VNemart Web service.
Input parameters:
- LookingFor
- FreeWord
- Country
- Area
- Employees
- Classification

Name: CTshow
Description of purpose: The web application of the Chamber Trade will call the Web service thought this interface in order to get the complete information about the selected enterprises.
Source of Input: Chamber Trade web application.
Input parameters: BusinessID

Name: VNshow
Description of purpose: The Web service of the VNemart will call the Web service thought this interface in order to get the complete Information about the selected enterprises.
Source of Input: VNemart web service.
Input parameters: BusinessID

Name: EnterCompetence
Description of purpose: The user will put new data about the competence that is going to be uploaded.
Source of Input: User.
Input parameters: competence database fields

Name: ValidateCompetence
Description of purpose: The administrator will accept or reject the information that the user gave.
Source of Input: Administrator.
Input parameters: Accept or Reject.

4.5.5.2 Functions

- **Search for companies**
  1. The petition comes from Chamber Trade web application
     1. Search companies in Chamber Trade database
     2. Translate categories from the ones in Chamber Trade to the ones in VNemart
     3. Send petition to VNemart Web service
     4. Integrate information from Chamber Trade and VNemart
     5. Present the information to the user
  2. The petition comes from VNemart Web Service
     1. Search Companies in Chamber Trade database
     2. Return results to VNemart Web Service

- **Show companies**
  1. The petition comes from Chamber Trade web application
     1. Looks if the company is from Chamber Trade or VNemart
     2. Returns the required information
  2. The petition comes from VNemart Web Service
     1. Returns the required information

- **Upload Competence**
  1. Receives parameters from the user
  2. Saves information in NewCompetence database

- **Validate Competence**
  1. Presents information from the upcoming competences
  2. Save information in Competence database

- **Search Competence**
  1. Search Competence according to the specified parameters
  2. Presents information about the competence to the user

- **Shows Competence**
  1. Returns complete information about the selected competences
4.5.6 Functionality Design

4.5.6.1 Search Companies in Chamber Trade

A function of the CT Web Service is to give search results according to the received parameters to the Web services that are subscribe to it. In this diagram we explain how the interaction between the web services and the CTDB (Chamber Trade Database) is done.

Figure 4.5.6-2 (State Chart Diagram Search Companies in Chamber Trade)

In the above diagram it is showed what the CT Web Service does when it receives a search request by the VNemart Web service.

4.5.7 Search Companies
A user through the web application of the Chamber Trade makes a search of companies according to certain parameters. The CTWeb Service look for those companies in the CTDatabase and then sends a request to the VNEmartWS. After that the CTWeb Service integrates the information and shows the result to the user.

The above diagram is the principal in the application, in this you can see the way the web service works and the way it performs its search around the databases available until it returns the result. Following the next diagram the user will be able to see and understand most of the functionality of the Web Service.
4.5.8 Return Companies

The user selects companies of interest in the CTwebpage and the CTWeb Service looks for the complete information of them in the CTDatabase and also sends a request to the VNemart web service. At the end the CTWeb Service integrates the information and send it back to the user.

The diagram above explains the primary steps that the CTWeb Service must do in order to send to the user the complete information of the selected companies.

4.5.9 Translator

The diagram above explains the primary steps that the CTWeb Service must do in order to send to the user the complete information of the selected companies.
The CTWeb Service has a translator component. And the above diagram explains the main steps of it.

### 4.5.10 Upload Competence

![Sequence Diagram Upload Competence](image)

Figure 4.5.10-1 (Sequence Diagram Upload Competence)

In this diagram is explained the way a User will register a new competence into the Chamber Trade Database using the Chamber Trade Webpage. This operation is registered in a table in the database before being approved by the Administrator (after the last the information must be saved in another table).

### 4.5.11 Validate Competence

![Sequence Diagram Validate Competence](image)

Figure 4.5.11-1 (Sequence Diagram Validate Competence)
In this diagram the Admin validates or not the competence and is advised to the user via email. The administrator does this operation using the Chamber Trade Webpage and accessing the Database with it in order to request, delete or store the competence in the database.

Figure 4.5.11-2 (Sequence Diagram Validates Companies Chamber Trade & VNemart)

This diagram works as a complement to the last one, this diagram shows what happens when the data introduced to the system by the User is not accepted by the Administrator.
5 Results

5.1 Implementation

5.1.1 Web Service

Our Web Service is called “Service”, and it is made by 3 main files that are: Service.asmx, Web.Config and Service.cs

5.1.1.1 Service.asmx

This file is the access point to outside applications. These applications in our case are the Chamber Trade Web application and the VNEmart Web Service.

In Service.asmx it is specified the language of the application (Language="C#"), the file that contains the source code (CodeBehind="~/App_Code/Service.cs"), and the class to invoke the source code (Class="Service").

Code of Service.asmx

```csharp
<%@ Web Service Language="C#" CodeBehind="~/App_Code/Service.cs" Class="Service" %>
```

Figure 5.1.1-1(Service.asmx)

5.1.1.2 Web.Config

This file specifies global configuration that is used by our Web Service. This file should be used if one has to change the value of some global variables such as connection strings to the database, authentication methods, how errors should be handle, etc.

For our project the most important feature on the Web.Config consist in the set of two variables. The first one has the connection string to the database and the second that has the destination of our supposed VNEmart Web Service.

Code of Web.Config

```xml
<configuration>
  <appSettings>
    <add key="connection" value="server=127.0.0.1; user = i_usr; password = mypass; database = mthesis;"/>
    <add key="vne1.Service" value="http://iesrv01.hj.se/vne/Service.asmx"/>
  </appSettings>
  <connectionStrings/>
  <system.web>
    .
  </system.web>
</configuration>
```

Figure 5.1.1-3(Web.Config)
5.1.1.3 Service.cs

This file contains the implementation of all the functions in our Web Service and it is the core element of our thesis. To develop it we first should include or import some functions and methods that are already implemented by C# and this ones are:

```csharp
using System;
using System.Web;
using System.Web.Services;
using System.Data;
using System.Data.SqlClient;
using System.Xml;
using System.Configuration;
using System.Web.Configuration;
```

![Figure 5.1.1-4 (Service.cs)](image)

Next we defined the namespace from the Web Service and its binding:

```csharp
[WebService(Namespace = "http://codeproject.com/WebServices/",
Description = "This is a demonstration WebService.")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
```

![Figure 5.1.1-5 (Service.cs)](image)

A web service is like any other class of C# with the difference that it has to be declared as a web service and to do that we did the following:

```csharp
```

![Figure 5.1.1-6 (Service.cs)](image)

Our solution has a constructor that has 2 methods in it.

```csharp
public Service () {
    dbLoad();
    DB();
}
```

![Figure 5.1.1-7 (Service.cs)](image)

dbLoad(): this method reads from the Web.Config the connection String of the database. DB(): this method opens the connection with the database.

We decided to put these two methods in the constructor, so the web service connects only once to the database and not every time it is called by some other application. “Service.cs” has some methods that can be invoked through the net. The first one of this is:

5.1.1.4 [WebMethod]

```csharp
public string search(string inquiries1, string freetext1, string CountryId1, string GeoArea1, string NoEmployeeId1, string ProductClassId1, string prueba)
```

![Figure 5.1.1-8 (Service.asmx)](image)
This method is called search. It is used by the Chamber Trade Web application and it returns some brief description of the specific inquiries that are located in the Chamber Trade and the VNEmart databases. The returned information is presented in form of a string and contains data from inquiries that fulfill the parameters given by the user.

To do all this, search invokes 3 methods:

1.1) private string createQuery()
1.2) private void exeQuery(string query)
1.3) private void translateClass()

1.1) private string createQuery(): is a method that creates a string that has an SQL statement. This SQL statement is created according to the parameters given by the user and it will be used later by exeQuery(string query).

1.2) private void exeQuery(string query): is a method that does two main tasks. The first one is to retrieve the records from the database. The second is to create HTML code that includes the results retrieved by the Database. This code is encapsulated in a string.

1.3) private void translateClass(): is a method that does 3 principal tasks:
   1.3.1) Translate the Chamber Trade Clasifications into the VNEmart Clasifications.
   1.3.2) Call the VNEmart Web Service.
   1.3.3) Create a string in the form of HTML code with the results brought by the VNEmart Web Service.

To do the first task there are two methods involved:

1.3.1.1) private string createQT()
1.3.1.2) private string executeQT(string q)

1.3.1.1) private string createQT() is a method that will create an SQL Statement encapsulated in a string, that will be used to retrieve the classifications from VNEmart according to the given classifications from the Chamber Trade. This is done by the use of a table that maps all classifications from Chamber Trade against all classifications of VNEmart.

1.3.1.2) private string executeQT(string q): is a method that executes the query created by createQT() and then writes the results in a string, where each classification is separated by a blank space: “ “.

1.3.2) The next task from the private void translateClass() method is done by: private void getWS(string classlist).

1.3.2.1) private void getWS(string classlist) is a method that calls the VNEmart Web Service and then gets the results that are encapsulated in an XML document. The format of this XML Document is called XMLSearch and is explained in a later section.

1.3.3) The last task from the private void translateClass() method is done by private void createResultVNE(XmlNode blabla).

1.3.3.1) private void createResultVNE(XmlNode blabla) this method creates HTML code with the values that are found in the XML file brought by the VNEmart Web Service.

5.1.1.5 [WebMethod]

| public string showResults(string id, string cmp) |

Figure 5.1.1-9 (Service.cs)

This is the second web method and it is called by the Chamber Trade Web Application and it returns the complete information from the selected inquiries that are in the Chamber Trade and the VNEmart Databases. This method returns a string that has the HTML code that will be shown by the Chamber Trade web application.
The parameters received are two strings. The first string is called id and has the unique identifiers selected by the user from companies that are in the Chamber Trade database. Each unique identifier is separated by " ". The second string is called cmp and it has the unique identifiers selected by the user from companies that are in the VNEmart database. Every unique identifier is separated by " ".

showResults invokes 3 different methods:

2.1) private string createQResults()
2.2) private void execResults( string q)
2.3) private void getVNECompanies(string companies)

2.1) private string createQResults() is a method that creates the SQL Statement that will be able to bring the complete results of the selected companies from the Chamber Trade database.

2.2) private void execResults( string q) is a method that does two main tasks. The first one executes the query given by createQResults and the second creates a string the HTML code that includes the results brought by the Chamber Trade database.

2.3) private void getVNECompanies(string companies): is a method that calls the VNEmart web service and ask for the complete information about the companies that are in the VNEmart database and have the unique identifiers given by the user. The results will be given in the form of an XML file(XMLResults). To take out the results of this file and translate them into HTML code, the method private void showvneCompanies(XmlNode bla) is invoked.

2.3.1) private void showvneCompanies(XmlNode bla): this method constructs HTML code according to the results that are in the XML file data structure.

### 5.1.1.6 [WebMethod]

```
public XmlNode searchCT(string inquiries1, string freetext1,
string CountryId1, string GeoArea1, string NoEmployeeId1, string
ProductClassId1, string prueba):
```

This method will be called by the VNEmart web service. It retrieves the specific companies that are located only in the Chamber Trade Database, and encapsulates the results in an XML file. The parameters freetext1 and ProductClassId1 might have several values, that should be separated by " ". This is done by the help of the following methods:

3.1) private string createQuery()
3.2) private XmlNode exeQueryCT(string query)

3.1) private string createQuery(): This method is already specified in 1.1
3.2) private XmlNode exeQueryCT(string query): This method receives a query string, executes it, and retrieve some specific information from the Chamber Trade Database. After that this method creates an XML Document(XMLSearch) that contains this specific information.

### 5.1.1.7 [WebMethod]

```
public XmlNode showResultsCT(string id):
```

This is the fourth web method of the web service and it is called by the VNEmart Web Service. This method returns the complete data from the selected inquiries that are in the Chamber Trade Database and encapulates it in an XML File.

The parameter received is a string called id and it has the unique identifiers of the companies whose information should be retrieved. Each unique identifier is separated by " ".

34
showResultsCT invokes 2 different methods:

4.1) private string createQResults()
4.2) private XmlNode execResultsCT(string query)

4.1) private string createQResults() is the same as 2.1
4.2) private XmlNode execResultsCT(string query): This method executes the query created by createQResults, and puts the results in an XML Document(XMLResults).

5.1.8 Format of XML Documents

XMLSearch
The format of XMLSearch is the following:

```xml
<?xml version="1.0"?>
<Document>
  <company>
    <BusinessID>(value)</BusinessID>
    <InquiryNumber>(value)</InquiryNumber>
    <Country>(value)</Country>
    <CompanyDesc>(value)</CompanyDesc>
    <RegDate>(value)</RegDate>
    <NoEmployee>(value)</NoEmployee>
  </company>
  ...
</Document>
```

Figure 5.1.10 (XML Search)

XMLResults
The format of XMLResults is the following:

```xml
<Document xmlns="">
  <company>
    <BusinessID>(value)</BusinessID>
    <InquiryNumber>(value)</InquiryNumber>
    <QuePriority>(value)</QuePriority>
    <RegDate>(value)</RegDate>
    <InquiryType>(value)</InquiryType>
    <BusinessInquiry>(value)</BusinessInquiry>
    <CompanyName>(value)</CompanyName>
    <Address>(value)</Address>
    <ZipCode>(value)</ZipCode>
    <City>(value)</City>
    <Country>(value)</Country>
    <GeoArea>(value)</GeoArea>
    <Phone>(value)</Phone>
    <Fax>(value)</Fax>
    <Email>(value)</Email>
    <Www />
    <StartYear>(value)</StartYear>
    <TurnoverUSD>(value)</TurnoverUSD>
    <NoEmployee>(value)</NoEmployee>
    <CompanyDesc>(value)</CompanyDesc>
    <ContactPerson>(value)</ContactPerson>
  </company>
</Document>
```
5.1.1.8.1 Using the Web Service

The client calls the web service and it is done by a Web Browser. For the implementation of this approach we need the Web Service.htc file. This file must be in the same directory as the page that will use the Web Service.

Code

We made access the Web Service through javascript:

```javascript
<script type="text/javascript" language="JavaScript">

We made function called init() that is called by
<b onload="init()"> in the HTML code.

```javascript
function init() {
    sve.useService("http://server/myServ/Service.asmx?WSDL","Service");
}
```

with this function we located our web service.

Then we retrieved the parameters using normal ASP code and translate them into javascript code.

```aspx
<%
Dim inq=request("InquiryType")
 .
response.write("var inq=" & inq & ","
%>
```
We called the web method and intercept the errors and finally we put the results in sve. That is the id of a div tag that is inside the body of the document: `<div id="sve" style="BEHAVIOR: url(Web Service.htc)"></div>`:

```javascript
idCallID = sve.Service.callService(myResults, "search", inq, search,country,geo,emp,prod);
  document.getElementById("hola").innerHTML="Loading... ";
}
function myResults(result){
  if(result.error){
    var xfaultcode = result.errorDetail.code;
    var xfaultstring = result.errorDetail.string;
    var xfaultsoap = result.errorDetail.raw;
    hola.innerText = xfaultcode + " " + xfaultstring + " " + xfaultsoap;
  } else{
    sve.innerHTML = result.value;
  }
}
</script>
```

### 5.1.2 Database Integration

There are several ways to classify and subs classify objects. Not only the classifications and sub-classifications might be different, also there can be different amount of levels for the sub-categories between the schemas. Different organizations and countries classify things according to their common needs. In order to integrate two databases that have different types of classifications a classification mapping between them has to be done. In our case we had to map the product classification between the Chamber trade and VNemart.

Our proposal to do this Classification Mapping has the following steps:

1) Get the list of categories and sub categories of each databases.
2) Find a data structure in which it is possible to organize both classification schemas.
3) Create mapping rules with the help of an expert.
4) Identify the different types of mappings.
5) Create a database schema that facilitates the data retrieval for the integration.
6) Create a table with the mapping rules.

**Get the list of categories and sub categories of each databases.** First of all to do any translation, mapping or integration it is necessary to know which data we are going to use in this process.

**Find a data structure in which it is possible to organize both classification schemas.** After retrieving the data required for this mapping, it is necessary to organize it in a way in which it is possible to use in the future. In the case from the business classification from Chamber Trade and VNemart we found that both data can be organized in the structure of a tree, where the Chamber Trade classification has two levels of depth and the VNemart has three, as it is shown in the following figure.
Create mapping rules with the help of an expert. At this point the help of an expert in the field is required in order to know which classifications from one database correspond to the classifications of the other database.

Identify the different types of mappings. After the expert shows which classification in one database corresponds to the other one in the other database we discover several types of mapping situations:

a) Classification from database A = Classification from database B. This is the perfect mapping situation both classifications are called the same and contained the same kind of data.

b) Classification from database A > Classification from database B. A specific classification from database A in level 1 contains all and even more sons that a classification of database B in level 1 has. For example Classification A in level 1 is called “electronics” and its sons are “CD players”, “television” and “MP3players”. And in Classification B is called “Electronic products” and has “television” and “CD players” only.

c) Classification from database A shares some elements with Classification from Database B. For example Classification A in level 1 is called “foodstuff” and it’s sons are “yogurt”, “cereal”, “cookies”; And Classification B in level 1 is called agriculture and has the following sons “yogurt”, “cereal” and “tractor”.

Create a database schema that facilitates the data retrieval for the integration. After analyzing how the data is going to be managed and how many different subcategories exist for each category, the approach to put all categories and subcategories in the same table was taken. The table has the name of the category and its unique identifier. To be able to select a classification with all its classifications each unique identifier has certain characteristics. In example, the first classification that is called “foodstuff” has the unique identifier “10000” and its sons have a unique identifier starting with “10001”. For the second category “electronics” the unique identifier starts with “11000” and so on. So the approach to select all elements from electronics is to use the SQL operator “like’11%’ in the where statement of the query.
Create a table with the mapping rules. The final step is to create a table with the following characteristics. The database has two fields, and this are the unique identifiers of the both table that have to be mapped. Each son from Classification A is going to be mapped against all sons from the Equivalent Classification B. An example of this is:

<table>
<thead>
<tr>
<th>Classification A</th>
<th>Classification B</th>
<th>Classification B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 1</td>
<td>FoodStuff</td>
<td>“Agriculture”</td>
</tr>
<tr>
<td>Sons</td>
<td>“yogurt” 10001</td>
<td>“yogurt” vn10001</td>
</tr>
<tr>
<td></td>
<td>“cereal” 10002</td>
<td>“cereal” vn10002</td>
</tr>
<tr>
<td></td>
<td>cookies” 10003</td>
<td>“tractor” vn10003</td>
</tr>
</tbody>
</table>

So the translation table will look like:

<table>
<thead>
<tr>
<th>Id Classification A</th>
<th>Id Classification B</th>
</tr>
</thead>
<tbody>
<tr>
<td>10001</td>
<td>vn10001</td>
</tr>
<tr>
<td>10002</td>
<td>vn10002</td>
</tr>
<tr>
<td>10003</td>
<td>vn24001</td>
</tr>
</tbody>
</table>

So if the user make a search on Classification A on Foodstuff the following query is going to be created:

“select distinct (Id Classification B) from translate where Id like’%10’”

And so the result is going to be what we want that is: vn10001, vn10002, vn24001.

This is the complete method that we used in our project to integrate two databases by mapping their classification table.

5.2 Hypothesis Testing

Intercepting Data with our Web Service:
In the figure 5.2-1 the data to be intercepted by the Web service will be:

- **Find Companies who are:** Looking for a co-operation partner
- **Enter free search word:** stuff we
- **Country:** Ukraine
- **Graphical Area:** -All-
- **No of Employees:** -All-
- **Classification of products/services:** electronics.

The figure 5.2-1 shows the retrieved results from the Web Service. The data received is the unique identifier from the requested value from a specific table. As an example, “Inquiry: 1” is displayed because of the previous selection (figure 1) “Cooperation Partner” and cached from the InquiryType Table.

From the figure 5.2-2 it is possible to observe that the web service captured: “stuff” “we” in the freetext field, this field is not included in a table due to the fact that is text written by the user. For Country the intercepted value is “7” that it is the unique identifier for “Ukraine” in the Countries table. (Figure 5.2-2b)

For “GeoArea” and “NoEmployeeId” the web service receives and displays the value “0”. The value 0 means “all the values”
For ProductClass we intercepted “1” that is the unique identifier for electronics in the table ProductClass.

For intercepting this data in our web service the following code is used:

```csharp
[WebMethod]
public string search(string inquiries1, string freetext1, string CountryId1, string GeoArea1, string NoEmployeeId1, string ProductClassId1, string prueba)
{
    inquiries = inquiries1;
    freetext = freetext1;
    CountryId = CountryId1;
    GeoArea = GeoArea1;
    NoEmployeeId = NoEmployeeId1;
    ProductClassId = ProductClassId1;
}
```

Identifying data:

The following image shows how the web service indentified the data and created the needed query.

Since the data is useless without interpretation, the web service had to identify the intercepted parameters in order to create the query for information retrieval. This is shown in the “where” segment of the query, where it is possible to see how the parameter “freetext” was identified and converted into two variables: “stuff” and “we”. (CompanyDesc like '%stuff%' or CompanyDesc like '%we%')
In this segment “InquiryTypeID=’1’, Business.CountryID=’7’ and Business_ProductClass.ProductClassID = ‘1’” it is shown as intercepted before.

The code to identify the free text and add it to the query was:

```csharp
char[] delimiterChars = { ' ', ',', '.', ':', '\t', '\n' };

words = freetext.Split(delimiterChars);

for each (string s in words)
{
    j++;
}

for each (string s in words)
{
    if (words[i].Equals("") && i == 0)
    {
        break;
    }
    if (i == 0)
    {
        query = query + " and ( ";
    }
    if (i == j - 1)
    {
        query = query + " CompanyDesc like'%' + words[i] + ";
    }
    else
    {
        query = query + " CompanyDesc like'%' + words[i] + ";
    }
    i++;
}
```

Integrating and showing data from different Databases:

For integrating data between different databases a translation among the fields is needed. For matching the fields a table called “Translate” is used. This table contains the unique identifiers of the classification tables from both databases.

For integrating data between different databases a translation among the fields is needed. For matching the fields a table called “Translate” is used. This table contains the unique identifiers of the classification tables from both databases.

Figure 5.2-3b

To get the specific result form the developed web service a query needs to be created. The following image shows the created query:
After all the data needed is generated, the developed Web Service sends a petition to second web service. The second web service retrieves the information from the second database and returns its results. The following picture shows the petition:

Figure 5.1.2-5 (Demonstration of the Translation)

In this picture we can find that classlist = vt01, this is the result we get after applying the translation.

After the petition, the second web service returns our web service the next XML Document:
The result brought by our database according to the query is:

![XML document by VNemart](image)

Figure 5.1.2-6 (XML document by VNemart)

The result of the integration from both databases is processed and displayed by our web Service. The following picture shows the result integration.

![Results displayed from the requested Query](image)

Figure 5.1.2-7 (Results displayed from the requested Query)

![Results by the Web Service](image)

Figure 5.1.2-8 (Results by the Web Service)
Results

If we take a look to the result brought by the query generated by our web service and the result brought by the second web service, we can observe that our web service integrates data from two different databases. For making the data understandable to the users the web service translate the data into html code, and then the web service the code is send to the webpage as an encapsulated string. The following code segment is used for the previously described task:

```csharp
for (int _i = 0; _i < BusinessID.Count; ++_i)
{
    resultado = resultado + "<table width='450' border='0'><tr><td  align='right'><font face='Verdana, MS Sans Serif, Arial, Helvetica' size='2'> " + BusinessID[_i].InnerText + "</font></td><td  align='right'><font face='Verdana, MS Sans Serif, Arial, Helvetica' size='2'>Inq. no:</font></td><td ><font face='Verdana, MS Sans Serif, Arial, Helvetica' size='2'> " + InquiryNumber[_i].InnerText + "</font></td><td  align='right'><font face='Verdana, MS Sans Serif, Arial, Helvetica'
size='2'>Country:</font></td><td ><font face='Verdana, MS Sans Serif, Arial, Helvetica' size='2'> " + Country[_i].InnerText + "</font></td></tr>";
    resultado = resultado + "<tr><td wwidth='40' align='right' valign='top'><font face='Verdana, MS Sans Serif, Arial, Helvetica' size='2'>" + NoEmployee[_i].InnerText + "</font></td><td width='75' valign='top' align='right'><font face='Verdana, MS Sans Serif, Arial, Helvetica' size='2'>Employees:</font></td><td wwidth='55' valign='top'><font face='Verdana, MS Sans Serif, Arial, Helvetica' size='2'>" + NoEmployee[_i].InnerText + "</font></td><td wwidth='65' valign='top' align='right'><font face='Verdana, MS Sans Serif, Arial, Helvetica' size='2'>Reg.date:</font></td><td width='*'>" + RegDate[_i].InnerText + "</font></td></tr>";
    resultado = resultado + "<tr><td width='40'>nbsp;</td><td valign='top' colspan='4' width='*'><font face='Verdana, MS Sans Serif, Arial, Helvetica' size='2'>" + CompanyDesc[_i].InnerText + "</font></td></tr></table><br><br><br>";
}
```

Figure 5.1.2-9 (Sample code of the WS integration)

For making this process transparent for the user, the developed web service shows only the information that the user needs. The following illustration shows this.
After the results obtained we can prove the value of our hypothesis: [“Web services can be used to create a component for intercepting, identifying, integrating and showing data from different databases while being requested through a web application as a transparent procedure for the user.”].

5.3 Best Practices

5.3.1 Installation

During the first stage of the project and as first requirement of the software, it was needed to have a specific set of software development tools configured and running in the computer to be used. In order to achieve this we needed to follow some steps that were more than just installing each tool into the computer. In our first try the installation was not done in the way it’s supposed to be done, but the situation was more complex than just reinstalling which is in fact a complex situation as we never had direct contact with the computer and the installations.

The computer we were working on was part of a system that is controlled by a system administrator, but we needed to solve this anyway. So we needed to find a solution. First we tried to find the solution with the tool’s manufacturers, but nothing was either in the documentation coming with the tools or in their websites. After getting deeper into the subject and finding on forums over the internet the same problem and many comments about the solution, we got into a cycle of installing and testing of course not directly but through the system administrator. After trying several times we found the correct way to do it and everything was set to start working.
5.3.2 Solution to Installation

In order to be able to install in the correct way both IIS and .NET framework there’s two steps needed to follow and that are not many references about this on the internet but in the other hand many people having problems with it.

1. First of all, Install IIS in the server where the application is going to run
2. Check that the server is able to run .asp
3. Finally install .NET framework into the server.

The last three steps worked in our case so probably this will also work in further projects facing the same problem.

5.3.3 Webconfig

There are steps to follow after installing the needed tools in the server. If the application needs to be shown on different computers than the server, it is important to make one modification to the Web.config file. Otherwise a problem of not being able to retrieve the information from other computer than the server will be faced. This error is a common error if you are working in the server (as you don’t have the chance quite often to see the application running from a client computer). The image below shows the error you might get if you forget to write that single line.

As you can see on the image, the error is really easy to overcome (as long as you read the information) but that doesn’t mean that it won’t be embarrassing to show it in a presentation in front of a client just by forgetting to write a line.
The solution to this problem is quite simple, is just by adding a new line setting off the custom errors between the `<system.web>` tag and the situation gets resolved.

In the next image is possible to see the solution for this error which looks like the marked line in the next image.

![Figure 5.3.3-2 (Correct settings of Web.config)](image)

5.3.4 DNS

The next “best practice recommendation” we can say is about the DNS. We had a problem where the DB and the web service were not able to be found, so we got some ideas about the solution. First we thought that the Data string connecting to the database was wrong, so we changed it pointing to local host and to the IP address to the local host (127.0.0.1), we also thought that the web service had a wrong configuration. We needed to find the way to solve it and we did. The answer we found after searching was quite easy but for this project was not in complete under our control.

In order to solve this last situation we needed to go back to the server administrator in order to configure the server to have a DNS. This configuration is something that will help you stop having problems with the services that you want to have both from the data base and from the web service. The solution is quite easy but as well is possible to forget about this when developing and have the same problem, which can become uncomfortable if you forget about setting the solution.

5.3.5 Compilation problem

Another situation we had to take care was the compilation of the files. This issue was something we found out on the run and by experimentation, why found that every time we wanted to compile a file on the .NET framework and run it after; we got an error message like the one in the image below.
Results

Figure 5.3.5-1 (Compilation error)

This error was a little difficult for us to solve it, but at the end we found out the way to overcome with it. And it’s simply to save the file instead of compiling it. Save the file and then try to use it. The file will work as normal and without a problem (at least that happened to us). In case you have already tried to compile your file, our suggestion is copy the complete file, paste it in a new file and save instead of compiling the new file. Now delete the old file. This is not a so elegant solution but we saw it works and has no problems at all.

5.3.6 Firewall

The last situation we think is important to consider when doing a project is related to the firewalls and the permissions of accessing services to the computer hosting the applications. When we developing this software, we got confused by not having access to the application, this made us think about the code and many other situations different than just the permission rights from to the server. The problem we faced was that our application was not able to access to the service running on the server when called from the same server.

Our advice is before starting to check on your code and on your work, in fact before starting to work on the project, is to go through the system and make sure that everyone has the rights needed to access without having problems. In case this is not possible to be done at the beginning is always good to keep in mind that some access errors can exist because of a bad configuration instead of bad coding.
6 Conclusion

In the realization of our thesis we show how it is possible to integrate heterogeneous databases and provide uniform access with the help of web services. Our approach proposes the use of two web services. Each one of them communicates with the database situated in their location, organizes the data extracted from that database and sends the information to the other web service.

With the use of SOA, in particular web services, a user is able to look for information that is situated in different databases. As we already proved in our realization.

There are many different methodologies to search data in databases with different schemas. At the moment there is no best method to do that. As we present in our research there is no automatic method to integrate data from databases that have different schemas. The need of an expert is still mandatory and this expert with the help of a developing team needs to select the best approach for each different project.

As a proved form our investigation data stored in different databases situated in different locations can be accessed using web services. To accomplish the data integration form different databases service oriented architecture can be used. Using SOA opens the possibility to enterprises for establishing partnerships and strategically fragmentations to share data situated in different locations requested from authorized users receiving the integrated information in a single view from a location free environment.

During the development of this project we found that software development has many things to be improved, it is true that software development is growing everyday and that new technologies are coming with easier standards and easier ways to be implemented. Sometimes it is fictitious that some of these technologies are completely friendly and easy. Some of the new tools for developing these technologies come into the market with a lack of documentation the information for answers regarding to problems can not be found straightforward trough specialized documentations.

Most of the times it is easy to create a completely new application from scratch than using the preconfigured tools and methods that the new development tools offer. Since the new tools are not documented efficiently. Using new technologies require a learning time, and this time has to also be considered under the development of a project where the technology is not mastered to the developer.

Web services are not as compatible and easy to use as they are marketed. Web service technology run in a special environment and setting this environment up brings a lot of constraints that have to be taken into consideration. There are certain steps and practices that must be followed in a specific order to make the system work properly.

Web Services bring several functionalities and are easy to create for new developments and for upgrading systems made with the new tools available in the market, in the other hand web services are not hundred percent compatible with older technologies. In these days many of the software development is about improving or adding functionalities to existing systems. If the existing software was developed with technologies that are not compatible with the web service, it becomes a challenge and many new and unexpected problems will arise. Working with these issues requires a lot of time and effort.

Before starting to develop web services an analysis of the running environment is recommended. That environment should be compatible with the web service technology that is going to be implemented; otherwise different approaches should be considered.
7 References


[10] Sarah Williams, Charlie Kindel The component object model: A technical Overview, MS Corp October 1994


8 Appendix

8.1 Time plan