2011

Automated Service Composition for Software Customization

Niruppama Amarnath
University of Windsor

AUTOMATED SERVICE COMPOSITION FOR SOFTWARE CUSTOMIZATION

By
Niruppama Amarnath

A Thesis
Submitted to the Faculty of Graduate Studies
Through the School of Computer Science
In Partial Fulfillment of the Requirements for
The Degree of Master of Science at the
University of Windsor

Windsor, Ontairo, Canada

2011

AUTOMATED SERVICE COMPOSITION FOR SOFTWARE CUSTOMIZATION

By

Niruppama Amarnath

APPROVED BY:

04 May, 2011

Author’s Declaration of Originality

Abstract

While Service-Oriented Architecture helps the realization of Software Product Line, the current practice of manual composition of service components has become a bottleneck for the automation of customized software production. The web service-based applications are increasing in the present days. The use of web services for the development of the software with the idea of product line is a new feature.

This approach not only uses clients' requirements for the identification of candidate services, but also takes sensitivity into consideration when deciding the most suitable ones for a particular custom-made software system. This enhancement of service selection and composition leads to the development of an enhanced framework of service integration life-cycle, whose operation is presented in a new algorithm in this thesis. In addition to details of the framework, a case study is also included to examine the process of candidate identification, sensitivity analysis, service selection, and system composition for a customized online shopping system.

Dedication

To my dear parents and sister for their encouragement and support of a lifetime

Acknowledgements

Words cannot express my gratitude and support from my supervisor Dr.Xiaobu Yuan. His constant encouragement and clear thoughts about explaining the research area helped me to fetch an insight and work ahead. During the thesis writing period, Dr.Yuan gave me excellent ideas which helped me to achieve things easily. Without his guidance, this work would have been impossible.

I would like to thank my internal and external reader Dr.Dan Wu and Dr.Gokul Bhandari for their valuable comments that helped me to improve my experiment and overall quality of the thesis.

I would like to take this opportunity to express my gratitude to my friend HariHaran VijayKumar for providing me valuable suggestions and also an industrial view about this work. A special thanks to Mohit Sud for providing me with the details on his area of research from which I was able to extend my work significantly.

Finally, I would like to thank all my friends who stood by me during my entire Master's program. This success could not have achieved without your support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Motivation

1.1 Introduction

Service Oriented Architecture (SOA) refers to the process that use internet to link software components. This involves service selection, composition, matching, designing and mapping. There are lots of available technologies. Based on technology used, interface description, protocol support and service registry is taken care accordingly.

A system follows its own technique for its construction of services, in this paper different approaches are studied and the best feasible method can be used. A comparative study of different approaches can also be done. After the service discovery, registration, and composition, quality of service need to be validated.

The validation of services will make the system give a better performance. The component based software development helps to plan the phases and support the modules. The framework is helpful to communicate between the different modules which help in reusability, maintenance and support.

Human computer interaction is based on the interaction between user and computer. The user will be specifying the requirement to the system. The system will choose the best web service for the development.

1.2 Web Service Composition

Web service composition involves the open, standard- based approach for connecting the web services together to produce the new business software. This composition helps to reduce the complexity, time and cost increasing the overall efficiency.

For the service composition a dynamic, flexible and adaptable framework is created. The framework should provide a clear understanding between the process of logic and the web services being involved. Web service dynamic composition involves many methods.

1.3 Structure of the Document

This document will be exploring the details of SOA which the recent technology used in the software industry. The following chapter will give an literature survey of SOA with the issues and drawbacks. The next chapter will involve the service selection, matching and composition. The different algorithms are being discussed with their advantages and disadvantages. Followed by a new proposed method is being introduced. An experiment is designed to evaluate the performance of the newly proposed algorithm and it is compared with the existing algorithms. The experiment demonstrates the efficient use of service composition which will reduce the time consumed for constructing new software.

Chapter 2

Review of Literature

2.1 Background

Service Oriented Architecture (SOA) is a software model in which automation logic is separated into smaller, distinct units of logic. Together, these units compose a larger piece of business automation logic. Individually, these units may be distributed. Each unit of logic is loosely coupled and may be reused across multiple applications as well as aiding in several different project objectives.

There are three important components in a SOA software model namely: Service Provider, Service Registry, and Service Requestor, as seen in Fig 2.1 [17].

- Service provider – capable of creating and publishing a service to a registry and makes it available through the Internet.
- Service requestor - performs service discovery operations on the service registry to find the needed service, then accesses that service.
- Service registry - aides service providers and requestors to find each other by acting as the registry of services.

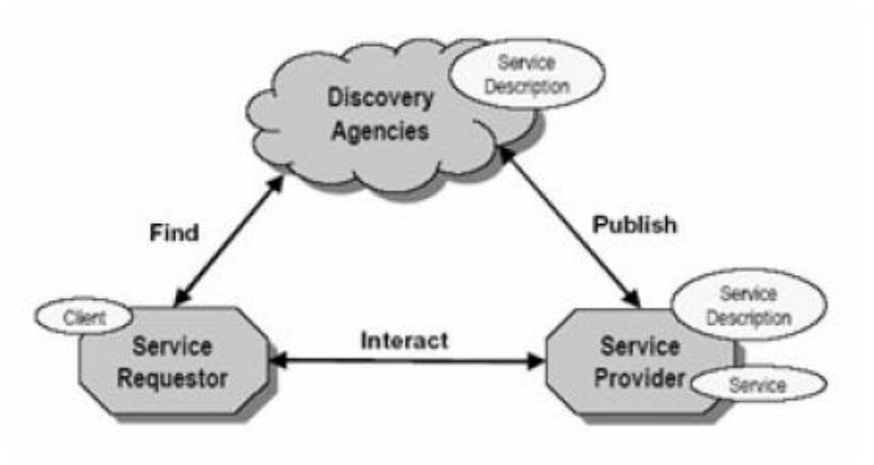


Figure 2.1 : Basic Components of SOA

Each unit must conform to a basic set of principles and standards that will allow them to grow independently while maintaining its ability to work conjointly with other units in logic. Service is designed based on their description, establishment, refining and adaptation to business processes. Though there has been no suitable method for development of service oriented business applications, focus is being made on the composition, design and granularity of services.

Services are defined as a single or multiple logic units which are encapsulated. It allows developers to utilize the unit of logic from within their own applications. Services are essentially then communication structure between applications and the unit of logic. The major advantage of SOA is, it reduces dependencies between customer and provider.

2.2 Overview on Web Services

Web services are one type of service provider to implement the SOA model. The web services are self contained, self describing web applications that are published, located, and dynamically invoked across the web. These support interoperable machine-to-machine communication over a network. The web service is a four layer model. The following figure is the structure of the model.

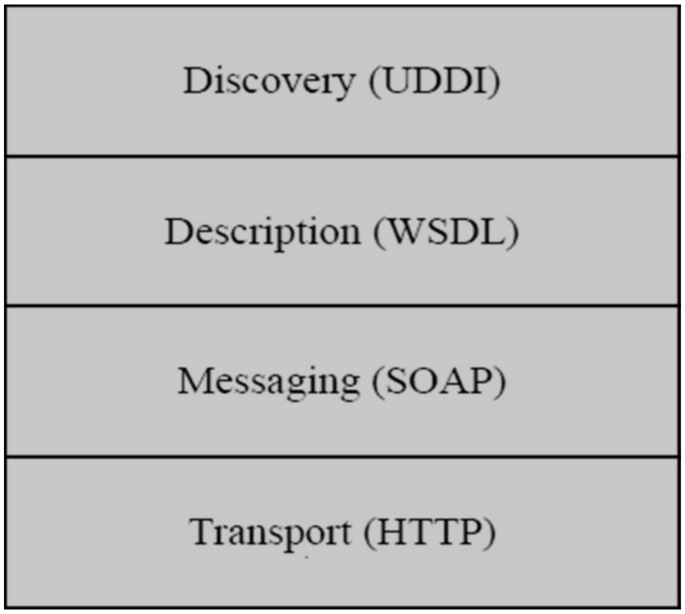


Figure 2.2 : Four layer model of Web Service Stack

Transport layer: The transport layer uses HTTP for passing the message between the network applications. This is the low- level protocol used for transporting between the internet and the model.

Messaging layer: The message layer uses XML format which can be understood by the recipient. SOAP(Simple Oriented Access Protocol) is common format for exchanging web service data over. HTTP.

Description layer: This is used to describe for a specific web service. Typically, WSDL(Web Service Definition Language) file is used to describe this information.

Discovery layer: This is a common registry. UDDI(Universal Description Discovery and Integration) is used for find the existing services.

Interoperability is achieved through a set of XML -based open standards, such as WSDL, SOAP, and UDDI. As there are only research work showing that UDDI can be used, there are no industrial usage, WSDL is being considered for the service composition case study.

2.3 Service Selection

Service selection is carried out based on the description of the services. Service description defines metadata that fully describes the characteristics of services that are developed on a network. This metadata is important and it is fundamental to achieving in loose coupling that is associated with an SOA. It provides an abstract definition of information that is necessary to deploy and interact with a service. The selected service has 3 different layers. They are entity service, task service and utility service.

2.4 Service-Oriented Design Method

The service-oriented design includes some steps and activities for classification. This is a five step activity like the general process model. The process steps are listed below with a brief idea about the action performed.[21]

By reviewing of the system layer, it can be observed that the services are layered based in their responsibility, reuse and multidimensional strategy. These layers are also analyzed.

Variation Analysis: As the next step in the design process, composition, interface and logic of the service component is created. Based on this the model is developed and the grouping is also done. Design phase service identification: Grouping of different services is the prerequisite for having an effective design. These services are controlled and positioned.

Review granularity: Granularity can be used in two ways: First, it is applied to the scope of the domain the entire service implements. Second, it is applied to the scope of the domain that each method within the interface implements.

Process modeling: The relationships with the business processes are modeled in this phase. This is done automatically using a toll as the manual mapping causes inefficient and inconsistency.

2.5 Services Composition

2.5.1 Overview

Service models help to establish fundamental contexts and functional boundary services by providing generic and predefined types. Service composition involves the development of customized services, often consisting of discovering, integrating, and executing existing services. It's not only about consuming

services, however, but also about providing services. This can be done in such a way that already existing services are orchestrated into one or more new services that fit better to your composite application.

There are 2 types of composition. Static composition designs the services before the development of software. Dynamic composition helps to design the services based on the user's input. For our requirement, we would be using static composition. This is used in semantic based service composition. Services are designed for reuse. A design structure must be created before implementation. Business process diagrams are refereed for the creation of new services. Before we develop a proper design structure, the features of the services must be kept in mind.[20]

- Services have well-chosen granularity
- Services are cohesive and complete
- Service encapsulate implementation details
- Services accommodate multiple invocation patterns
- Services uses interfaces
- Modeling of services are done using state-transactions

Keeping process diagram is in mind, services are designed for the proposed idea. Developer must provide the information for following. [20]

- Responsibility of the service

- Rules of the service
- Method used for implementation
- No duplication of services

Conditions to apply a service, which means if a service has specific business process, it must be described and included in the service design

- Actors involved in the use of service
- Order at which the service is called
- Acknowledgement for the service invoked

Apart from this, the user must also remember the technical aspects of designing the service. Some of them are:

- The format in which the service is used – XML, SOAP or any other mechanism
- Protocol used for communicating with the service – HTTP or RPC protocols
- Availability of the service
- Expected acknowledgement for the service sent

SOA is both scalable and reliable. SOA simplifies the development and is capable and using different services and process for implementation. The developers must have domain knowledge and
basic programming skills. There are numerous ways to attain our goal. For our requirement, we would follow the given ways.

- Declarative techniques
- Abstractions
- Code generation
- Tooling
- Model driven development

2.5.2 Principles and Techniques

SOA provided a set of principles, techniques and theories for building a business process. The services are composed of activities (1) connection combined with data and control flows and (2) described with entry/exit conditions. Each activity of the service has a goal. This is needed for application development. [8]

Initially, services used components as web service. This was done based on inserting the SOAP/WSDL/UDDI layer over the existing software application or components that makes the web service possible to work. [8]

For development of service oriented application emerging technologies like J2EE, .Net, SOAP, UDDI and many others are used. These help to compose existing services into large functional units. For service composition, 2 aspects are being considered. [3]

(1) Designing the service composition automatically and integrate them accurately. Mostly the services are described based on their semantic knowledge ie using description language and registered in the UDDI/SOAP layer. Though the market has many description languages, we generally support only one description language. [3]

(2) This helps to ensure successful execution of service composition in dynamic environment. This composition is determined in the design phase. [3]

2.5.3 Web service Composition

Web services are self contained, modular units of application logic that can be used in business functionality to provide connections for other applications using a server and client or internet connection. Web services are being described using the WSDL (Web Service Definition Language) which is an XML based language. Web service composition must provide dynamic, flexible and adaptable framework. The framework must be able to separate between the process logic and web services being used. Further the framework must be able to compose higher-level services for new or modification of requirements.

2.6 Advantages

SOA is an erupting technology and there are many advantages of using this technology in any implementation. These are beneficial in the information technology industry. Some of the advantages are being listed below.

- Since the SOA is being implemented in a network, its application can be accessed anywhere on the World Wide Web as it is known for its independence.
- Reusability is high in SOA as there are loosely based services coupled together to perform one single function or an application.
- An SOA application is completely platform and language independent.
- SOA and web services are interoperable as they are on web and internet scale computing.
- SOA bridges between incompatible technologies.
- As SOA enables the feature of reuse, efficiency of producing new application is easier and simpler
- Simpler systems: As SOA is based on industry standards, complexity is reduces and becomes easier to get solutions.
- Lowering maintenance costs: Simplicity and easy maintenance reduces the costs.
- Enhancing architectural flexibility : SOA supports the composite solutions. This helps to interface with multiple systems using the simpler user interface.

Chapter 3

Service selection

As SOA has become popular in the industrial communities in the recent decade, the primary concern is to find the services that satisfy the performance and the requirement. The service sensitivity provides the developers to find the appropriate software service and its consistency and optimality are being checked. The approach focus on the technique of reusing the same service for different objectives. Several software services are available that helps to achieve the same desired output. Service requestors have a choice as to which service they can choose to incorporate within their application. Although these services aim to achieve the same output, they may produce different results for the same input parameters. Therefore keeping this in mind, a software service must be assured that they match the requirement. Service sensitivity is an useful tool in the model of construction and evaluation.

3.1 Product Line

Software product line practice has been found to reduce both the cost of the software and the number of defects. This is being considered for the development of service oriented architectures (SOA) systems. The idea of using semantic web service description language to encode the product configurations and using it as the input for the proposed framework helps to achieve a real-world application. Customizing the web services based on the functional and non-functional requirements to achieve the product line engineering approach is being discussed in the paper written by Hongyu, Robyn R. Lut and Samik Basu [21].

To manage the web services mechanism it should be remembered that the software product line should be satisfied. The conditions are stated in the form of business foals set by the systems for the accomplishment of a prescribed task.

3.2 Service Optimization

Selection of an appropriate web service for a particular task is a difficult challenge as the service repository will have similar functionality services a lot. This will involve both the functional and non-functional properties of the service. The functional property is referred what a service can do and the non-functional property refers to how a service can do it. [23]

For the developers creating a SOA based application that is dependent on the services and its composition, service optimization is very important. The basic idea is to ensure that they selected service will help to perform accurately matching the scope of the project.[22]

Developers can run the services to test their performance and the variation of the same functionality service are being compared using a comparator operator to identify the optimal service. The introduction of the service selection helps to the development better. When a service provider is creating new software, the main requirements are whether the requirement uses only one service or 2-3 services. Added to this, another main concern is that there may be services with same functionality.[24]

3.3 Web service discovery

Discovery of the web service means the activity of locating a web service that meets a certain functional criteria. Selection involves the activity of evaluating and the ranking of the discovered web services. The service selection is done based on the web service descriptions which are in ontological concepts.

The algorithm is based on the service sensitivity analysis. The main focus on this method is that if there are one or more web services which matches the requirement, the best service is being chosen. This module is a previous work of Mohit Sud and Dr.Yuan. The main idea behind this work is to find an optimal service for the given requirement. [25]

3.4 Semantic Matcher

A knowledge base is needed for doing the service matching and service selection. The ontology inference engine has the web service description and the request description. Different parameters are to be matched to obtain the exact service. The most appropriate service is being selected from the list of potential services available.

The service is being selected using the onotolgy technique. Rest is not a standard, or language or protocol but is used as an architectural base to work with the web and protocols technologies (eg: HTTP and XML). REST is a simpler than SOAP. REST helps to design the software better. It helps to create a server and client. This is the greatest advantage in the web service technologies making everything the same. [10]

A server and client component is necessary. The server component is responsible for interacting with the clients. The server will be the repository here having all the potential web services. The number of clients can increase depending on the gain of popularity of the application. A service provider can create a new service and can allow the requestors to access them for their applications. It is not possible that all the services would be unique. The result in selecting a service depends on the objective of the application. It is also required that the service requestor selects the most apt service for the given condition or else the application will be ruined.

The main objective of this thesis is to address the web service composition. As the web service composition involves the web service selection also, it is been given some importance.

3.5 Approaches to Sensitivity Analysis

There have been multiple approaches that are similar in nature to sensitivity analysis as it applies to various models, and are not restricted to the software domain. These approaches have not been applied to the SOA model as a means of effectively evaluating the performance of a service for the fulfillment of quality assurances and service selection in an effective manner.[11]

Brute Force -Works only on small models that take a short amount of time to solve, change the initial data and solve the model again to see what results you'd get. Classical Sensitivity Analysis - Applies to very large models that take a large amount of time to solve. The classical sensitivity method relies on the relationship between the initial table and any later table to quickly update the optimum solution when changes are made to the coefficients of the original table.

Computer Based Ranging - Simple information about how certain coefficients can change before the current optimum solution is fundamentally changed. John W. Chinneck, a professor at Carlton University, has done considerable research in regards to using sensitivity analysis to identify which data has the most significant impact on results in a software system. He used a combination of linear programming and computer based ranging to establish this information.

Analysis of Variance (ANOVA) - The ANOVA approach is very common in statistics. It involves performing computations to determine if the mean of two given treatments are equivalent or not. Although this approach has proven successfully in evaluating various statistical models, its methods are best suited for two factor based sensitivity analysis [11].

Design of Experiment (DOE) - DOE experiments are not restricted to software systems and may be performed on a variety of things including; software systems, people, plants, animals, etc. DOE allows for observation and judgment on the significance to the output of input variables that may be acting alone, or in combination with one another. DOE may be considered to be Sensitivity Analysis earliest ancestry. Its approach has not been adapted to software service [8].

3.6 Previous Research

Sensitivity analysis is a practical optimization mechanic that has been used to establish confidence and performance optimization in many things including statistical research and software applications. There currently has not been a sufficient amount of research aimed in regards to applying sensitivity analysis to the success and improvement of SOA. The most identifiable research has been conducted at the

University of Windsor with Dr. Xiaobu Yuan and a few former students [3]. Shangwei Duan, Tony Huang, and Dr. Xiaobu Yuan, University of Windsor, have demonstrated a technique that allows for two-factor based sensitivity analysis of SOA based services. There technique applies sensitivity analysis to SOA in an effort to evaluate and predicate software quality, as well as to identify optimal configurations in software services with only two factors [19]. Chunjiao Ji and Dr. Xiaobu Yuan, University of Windsor, have outlined a methodology for multi factor based sensitivity analysis of SOA based systems. Their approach is used to identify which if any individual factors or joint factors have significant effects on the performance of a software system [20].

Chapter 4

Different Approaches involved in Service Composition

There are different approaches for automatic composition of services. Some of the approaches are being discussed below.

4.1 Different Approaches of Service Composition

4.1.1 Process based

To perform a simple task, everyone will have a certain plan to execute. For example, if a professor is offering a course during “Fall 2010”. He will first let the department know the course syllabus and whether it is under graduate or master student course. During course registration, the department will let know the students about the availability of the course. Further, the professor will decide how many students he wishes to have for the course. Thus, everything takes place as per the process or plan. This is known as process synthesis. This is known more as static composition. For smaller problem this might work well. But for automatic web service composition, this lacks flexibility and scalability. User requirements are captured for automatic web service composition, encapsulation of variable and dynamic activities of the domain. Semantic and Ontology is used for describing the rules.

According to reference [8], for dynamic discovery of services, selection and composition will involve process ontology. This will involve rule set, choice relation, exclusive relation, prior relation, unordered relation and parallel composition. This is based on 3 tuples N (set of nodes), E(set of edges) and R(set of rules). These are extracted as relations and rules which compose an ontology tree.

As the next step, upon service request, system searches for target node in the process ontology tree and validates their interrelation before composing. To perform this action, the authors have introduces two more definitions node depth and parent node, based on which two algorithms are written. [8] This process based cannot be compared with any other approach. This is basically a search done using user specification involving the services capabilities and behavior. Since it is a workflow like technique, it can compose business process and integrate dynamically.

4.1.2 Graph based search algorithm

The service composition basically aims to decrease human intervention during service discovery, matching, ranking, filtering and reasoning. According to Mazen Malek Shiaa, Jan Ove Fladmark, Benoit Thiell used the graph based search algorithm for discovery and matchmaking based on the semantic annotation of the services. This helps to find all the possible combination for reasoning and validating the services.

The semantics of the services and their properties are captured for the composition of services. The description languages that are capable for capturing semantics are OWL-S, WSDL, and RDF. These provide information about properties of services. For instance OWL-S will have 2 parameters about the semantics of service. They are Functional (eg, input, output, conditions) and Non-Functional (QoS).This helps in service discovery. Service matching is an important issue for composition of services. [5] The graph-based search algorithm is based on modified breath-first search algorithm. With reference to [10], the algorithm is described. A tree-like structure is to be formed for the composition of the services. This will have a node which is a unique index, an edge connected the two nodes with the output from one node as input to another node, and branch and composition are sub-graphs of the overall graph containing distinct set of nodes and edges.

The initial phase will be discovery of services and matching the services based on semantic properties. Validation is done to ensure that the system is being built correctly. Later prototyping is carried out to implement this algorithm. The prototype of implementation will look like the below figure. [10]

4.1.3 Evolutionary algorithm

Based on the idea proposed by Thomas Fischer, Fedor Bakalov, Andreas Nauerz, Martin Welsch in their paper [7], mashups and evolutionary algorithms are discussed. Mashups provides programming language the model using which different web applications can be created. An appropriate mashup framework will allow user to combination of web services for development. Based on this idea, information is made available through the user's model for which the evolutionary algorithm is applied to make the existing application to gather information about web services.[7]

Creating a simple mashup requires the user's technical knowledge. Similarly mashup has certain disadvantages, but still it provides good aggregation with the data sources. It is important to achieve automatic and adaptive mashups. [7] WSDL is used for capturing the messages, endpoints and interfaces for the services. The semantic knowledge is captured by OWL-S. Thus, this helps in the parallel execution of web service composition.[7]

The system architecture is a framework, which tells the flow of information content between different modules. Knowledge representation and ontologies are important for mashups. This is used for combination of data from different sources. The information on web is represented using RDF (Resource Description Framework). The evolutionary algorithm is used in the composition module.[2] The evolutionary process is flows as shown in the diagram. The process starts with obtaining the possible set of inputs. The best combination is being determined. This way, the web service composition is planned, represented and optimized. The main objectives such as correctness, completeness and length are weighted by a specific linear combination are handled in this. These functions are calculated by means of different functions. [2]Therefore, this approach helps to test the scalability and correctness.

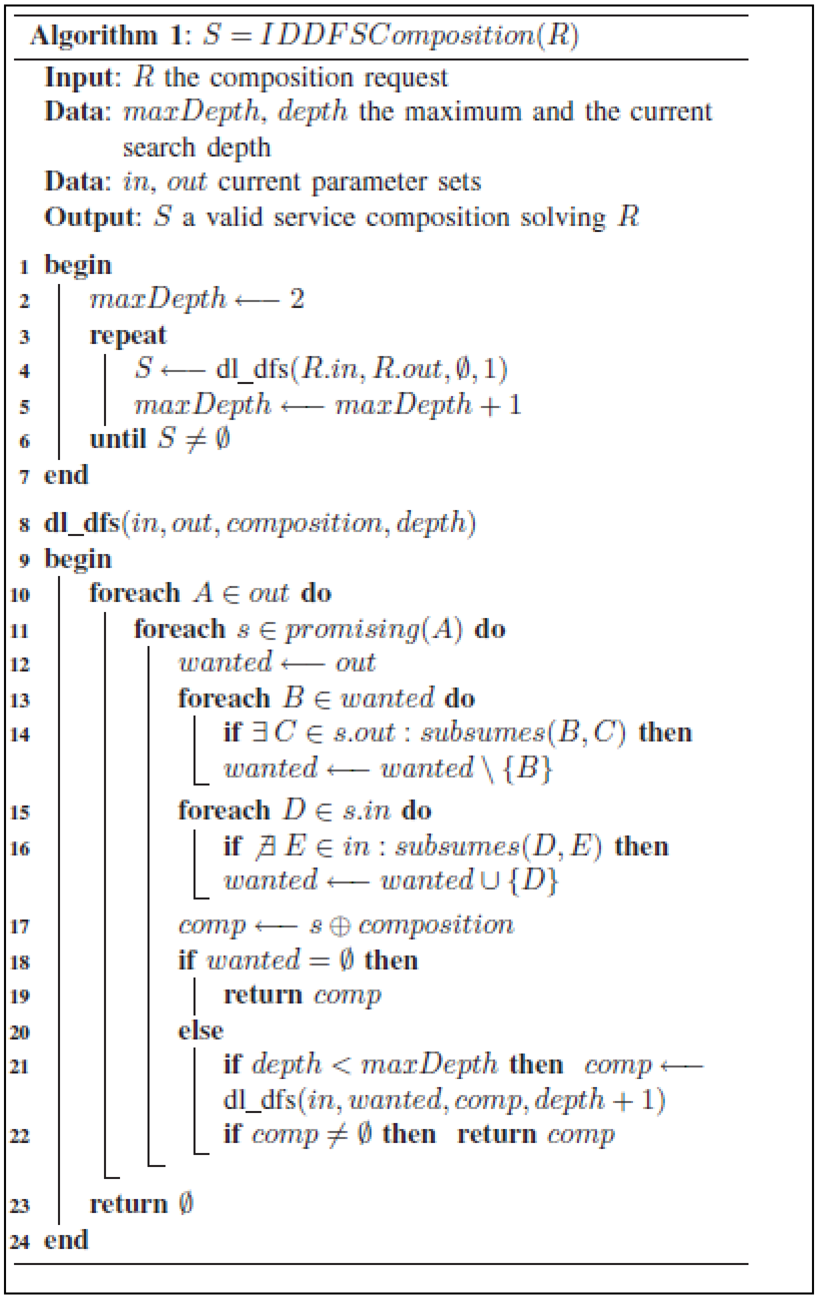
4.1.4 Feature based composition

Feature based composition helps to identify the selected features for deriving the result. It is based on the patterns/ models. This is top down approach ie the end result or the software product is first obtained and from there the set of features are being classifies. It considers 2 main phases. (1) feature and aspect modeling concerning to the language used (2) constraint analysis dealing with the feature and aspect level (3) model composition which tells about the process to derive a single product. [2]

This method does not explore the reusability and also dynamic feature selection for our web service composition.

4.1.5 IDDFS algorithm (Iterative Deepening Depth-First Search)

The authors Weise, Steffen, Comes and Kurt have discussed the following three algorithms in their paper “Different Approaches to Semantic Web Service Composition” which was submitted in the third international conference on internet and web applications held in 2008. This algorithm is a straightforward approach for composing the web services using the uniformed search. The search algorithm does not use any special mathematical formulas for the achievement of the result. It uses the iterative depth-first search. This helps to find the solutions quickly when the service repository is mall and tries to provide optimal solution if the problem requires exhaustive search.



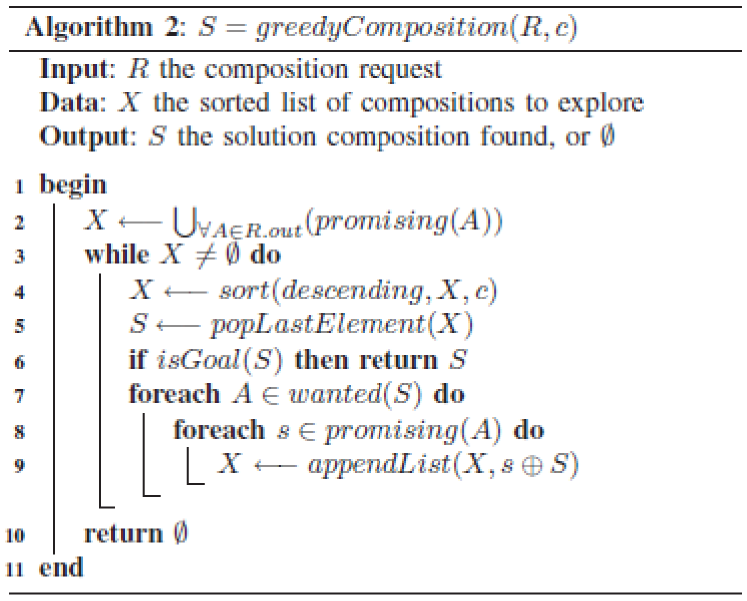
```
Algorithm 1: $S = IDDFSComposition(R)$
Input: $R$ the composition request
Data: $maxDepth$, $depth$ the maximum and the current
      search depth
Data: $in$, $out$ current parameter sets
Output: $S$ a valid service composition solving $R$

begin
    $maxDepth \longleftarrow 2$
    repeat
        $S \longleftarrow$ dl_dfs$(R.in, R.out, \emptyset, 1)$
        $maxDepth \longleftarrow maxDepth + 1$
    until $S \neq \emptyset$
end

dl_dfs$(in, out, composition, depth)$
begin
    foreach $A \in out$ do
        foreach $s \in promising(A)$ do
            $wanted \longleftarrow out$
            foreach $B \in wanted$ do
                if $\exists\, C \in s.out : subsumes(B, C)$ then
                    $wanted \longleftarrow wanted \setminus \{B\}$
            foreach $D \in s.in$ do
                if $\nexists\, E \in in : subsumes(D, E)$ then
                    $wanted \longleftarrow wanted \cup \{D\}$
            $comp \longleftarrow s \oplus composition$
            if $wanted = \emptyset$ then
                return $comp$
            else
                if $depth < maxDepth$ then $comp \longleftarrow$
                dl_dfs$(in, wanted, comp, depth + 1)$
                if $comp \neq \emptyset$ then return $comp$
    return $\emptyset$
end
```

Figure 4.1: Algorithm of IDDFS

According to the algorithm, web service composition is being built based on the valid web services. The loop is invoked iteratively increasing the depth limit at each step to provide a valid solution. The disadvantage of this approach is that it is slow for large memory repository. Apart from this it does not utilize any additional information for its search criteria. The efficiency can be improved remarkably if these disadvantages are being overcome.

4.1.6 Greedy algorithm

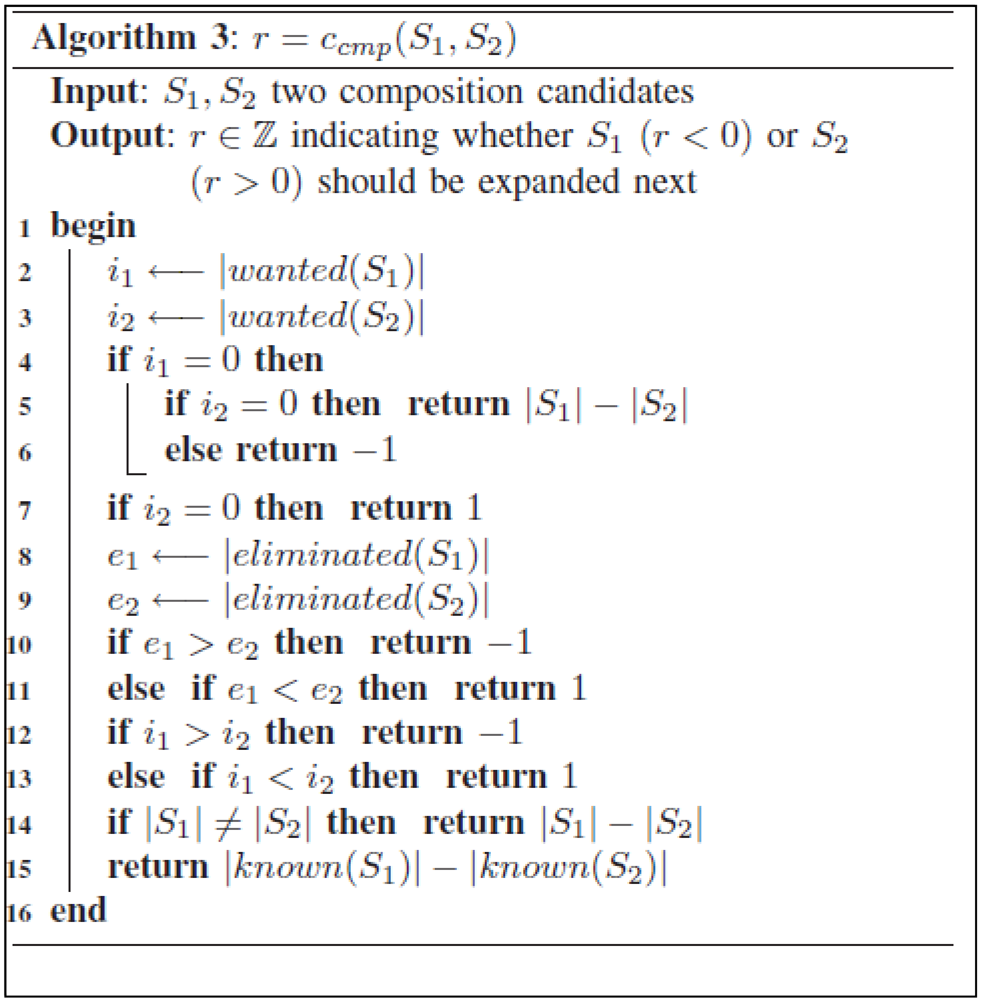
Greedy algorithm does the sorting internally using the comparator function. Based on the requirement, two best services are being short listed. The services are compared using the function. The best services is being validated based on the satisfying parameters. Mostly the service with shorter parameters is taken as the composition will become small and short.



```
**Algorithm 2**: $S = greedyComposition(R, c)$
**Input**: $R$ the composition request
**Data**: $X$ the sorted list of compositions to explore
**Output**: $S$ the solution composition found, or $\emptyset$
begin
    $X \longleftarrow \bigcup_{\forall A \in R.out}(promising(A))$
    while $X \neq \emptyset$ do
        $X \longleftarrow sort(descending, X, c)$
        $S \longleftarrow popLastElement(X)$
        if $isGoal(S)$ then return $S$
        foreach $A \in wanted(S)$ do
            foreach $s \in promising(A)$ do
                $X \longleftarrow appendList(X, s \oplus S)$
    return $\emptyset$
end
```

Figure 4.2: Greedy Algorithm 1

The comparator function involves the following steps.



```
Algorithm 3: $r = c_{cmp}(S_1, S_2)$
Input: $S_1, S_2$ two composition candidates
Output: $r \in \mathbb{Z}$ indicating whether $S_1$ $(r < 0)$ or $S_2$
        $(r > 0)$ should be expanded next
begin
    $i_1 \longleftarrow |wanted(S_1)|$
    $i_2 \longleftarrow |wanted(S_2)|$
    if $i_1 = 0$ then
        if $i_2 = 0$ then return $|S_1| - |S_2|$
        else return $-1$
    if $i_2 = 0$ then return $1$
    $e_1 \longleftarrow |eliminated(S_1)|$
    $e_2 \longleftarrow |eliminated(S_2)|$
    if $e_1 > e_2$ then return $-1$
    else if $e_1 < e_2$ then return $1$
    if $i_1 > i_2$ then return $-1$
    else if $i_1 < i_2$ then return $1$
    if $|S_1| \neq |S_2|$ then return $|S_1| - |S_2|$
    return $|known(S_1)| - |known(S_2)|$
end
```

Figure 4.3: Greedy Algorithm 2

4.1.7 Multi objective genetic algorithm

This is a genetic approach for solving the meta-heuristic optimization algorithms. It has the biology terms like mutation, crossover, natural selection and survival of the fittest. Comparative to the other approaches discussed before this has the advantage of optimizing the solution from the few assumptions made.

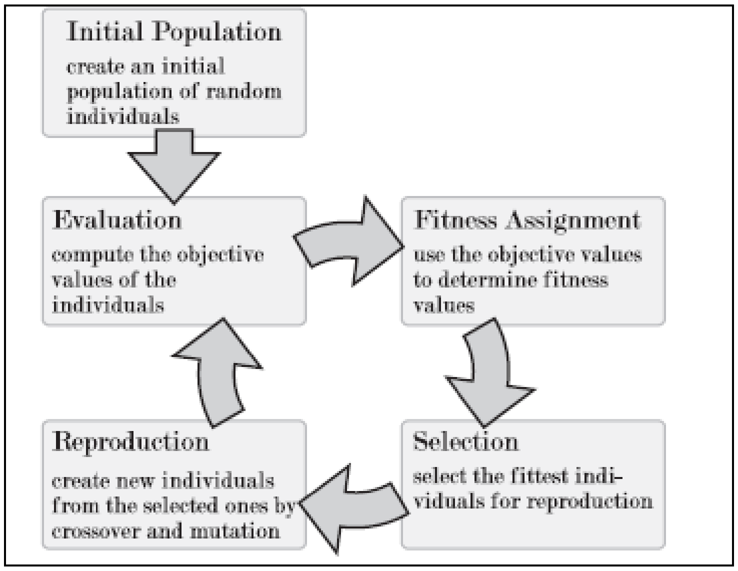

Initial Population
create an initial population of random individuals
Evaluation
compute the objective values of the individuals
Fitness Assignment
use the objective values to determine fitness values
Selection
select the fittest individuals for reproduction
Reproduction
create new individuals from the selected ones by crossover and mutation

Figure 4.4: Multi Objective Evolutionary Algorithm

The evolutionary algorithm involves some steps in it process of service composition. As the first step, all the services are being created. All these services are being tested and the best evaluated service is being picked for the composition.[4]

4.1.8 Bottom-Up Approach

The authors Muhammad Ahtisham Aslam, Jun Shen, Soren Auer, Michael Hermann have mentioned in their paper “An Integration Life Cycle for Semantic Web Services Composition”, about the following different approaches.

The bottom-up approach integrates the web technologies for the web service composition. This considers BPEL(Business Process Execution Language) for its integration. The basic view of this approach is that it uses the machine understandable descriptions of required services within process to realize the semantic descriptions. This helps to dynamically discover the required services. If a single service do not meet the requirement, recursive back-chaining algorithm is used to find the service that can be invoked. The main disadvantage of this approach is that it does not consider the pre and post conditions for the service discovery and composition.[4]

4.1.9 METEOR-S Approach

A dynamic tool is being developed for the web composition. This allows the designers to design a process based on the process and business constraints. The concept in which this works is that, it the service specification is being given from which the discover of services is being done. This is an abstract process. More of the functional specifications is being used for the service selection. The designers uses the BPEL for process modeling. Major downside of this approach is that manually selection of service is done for service composition. [4]

4.1.10 Template Based Composition: An AI approach

Artificial Intelligence can be used in any field. According to this approach abstract activities can be used to describe the service that is being used. Services are discovered based on their activities. The main idea of this approach is that it tries to focus on the value of preferences. This makes the services with high suitable specification to be ranked one in the bundle of services available. In [5] Evren Sirin, mentions that this will be using the Semantic Web technology (OWL) for writing the template specifications for the services. The AI planning is used for the discovery of services.[4]

4.1.11 WSMO Composition Approach

WSMO communications also developed a tool on dynamic composition for web services. The composition tool allows the developers to mention about their goals, mediators and control flow. The process involves the list of goals from which the specific gols is being taken into account. The type mismatch between the input and outputs are maintained by the mediators. This also uses the OWL-S (semantic web technology)[4]

4.1.12 Automated Composition by Using SHOP2

This approaches describes about how an AI planning system can be used for the web service description to compose the web services automatically. Though this approach deals with the planning, it only supports partially for the composing of the web services. At the same time, this also does not support composite process with all the OWL-S supported constructs. [4]

4.1.13 SWORD

This method is stated to be providing a set of tools for composition of web services. The concept of rule-based expert system helps to determine the automatic creation of composite services using the existing services. The negative aspect of this method is that, its idea is being limited to selecting web services for composition based on the input and output and also certain pre-conditions that will effect.[4]

4.1.14 Plaengine

This is a software system that provides the planning for the composition of services. This uses the meta-model approach. Meta-model here means that this will be uses the patterns for the service discovery. There are two main components involved: composer and enactor. The composer is liable for generating the composition of search planning algorithm to perform the action. The enactor is in charge of scheduling and execution of the services within a composition.[4]

4.2 Limitations of the approaches

There are major challenges on the existing approaches which is being summarized and given in a tabular column below. Some of the main issues of the dynamic composition are:

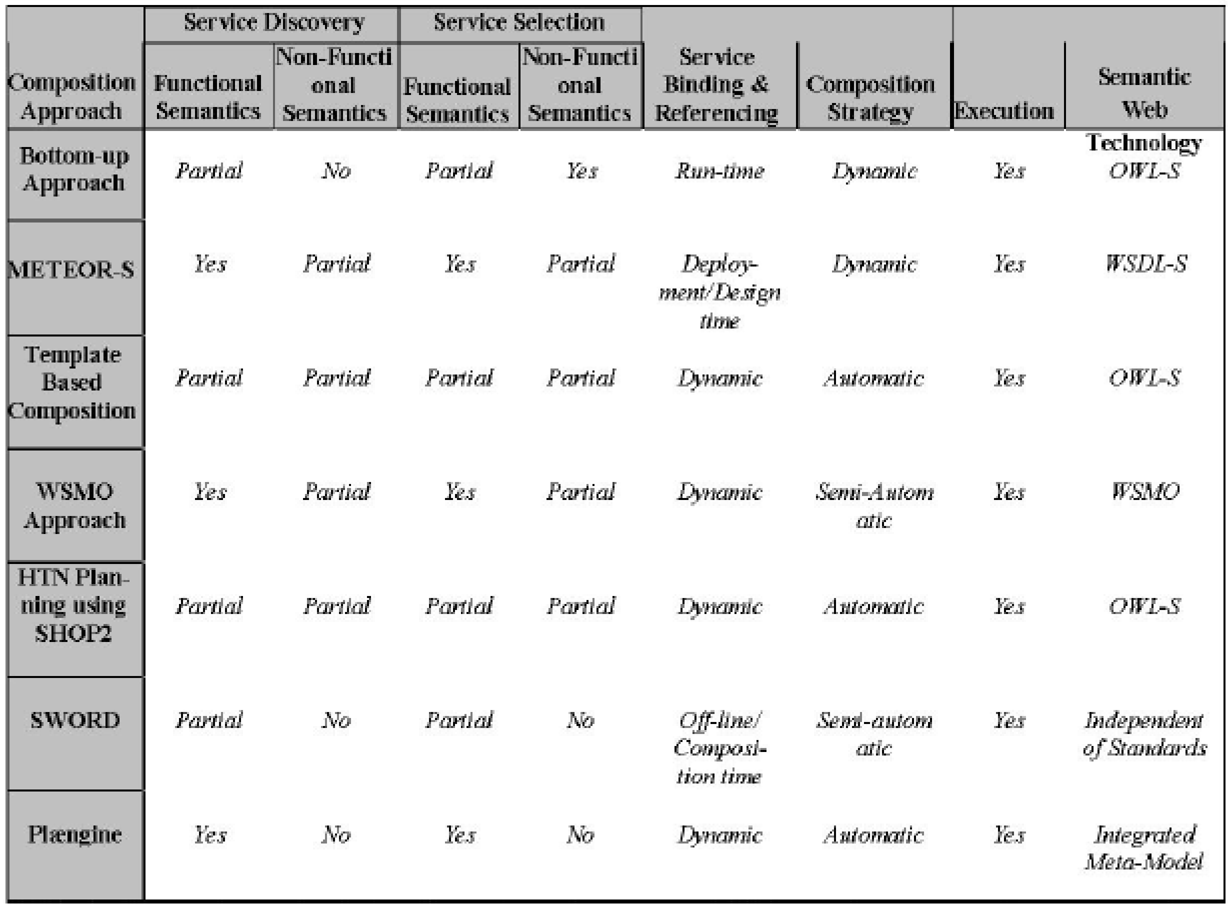
Service discovery and selection on the basis of matching functional and non-functional semantics: This deals with the basic functional semantics ie the input, output, pre and post conditions. The non-functional semantics is about the extra information (the currency rate, postal code etc). This is more concerned about selection of one service from a bundle of semantic services.

Service binding and referencing: How a service is being selected for the composition is being discussed here. Basically the workflow and AI planning is being described here.

Composition strategy: This involves the composition of the semantic web services. From the planning the final composition is being found whether it can be done automatic or semi-automatic.

Execution: The final output of the execution is being focused here.

Semantic web technology: The web service technology that is concerned for this approach is being discussed.



Composition Approach	Service Discovery		Service Selection		Service Binding & Referencing	Composition Strategy	Execution	Semantic Web Technology
	Functional Semantics	Non-Functional Semantics	Functional Semantics	Non-Functional Semantics				
Bottom-up Approach	*Partial*	*No*	*Partial*	*Yes*	*Run-time*	*Dynamic*	*Yes*	*OWL-S*
METEOR-S	*Yes*	*Partial*	*Yes*	*Partial*	*Deploy-ment/Design time*	*Dynamic*	*Yes*	*WSDL-S*
Template Based Composition	*Partial*	*Partial*	*Partial*	*Partial*	*Dynamic*	*Automatic*	*Yes*	*OWL-S*
WSMO Approach	*Yes*	*Partial*	*Yes*	*Partial*	*Dynamic*	*Semi-Autom atic*	*Yes*	*WSMO*
HTN Plan-ning using SHOP2	*Partial*	*Partial*	*Partial*	*Partial*	*Dynamic*	*Automatic*	*Yes*	*OWL-S*
SWORD	*Partial*	*No*	*Partial*	*No*	*Off-line/ Composi-tion time*	*Semi-autom atic*	*Yes*	*Independent of Standards*
Plængine	*Yes*	*No*	*Yes*	*No*	*Dynamic*	*Automatic*	*Yes*	*Integrated Meta-Model*

Figure 3.5: Limitations of the different approaches

A successful model of selection and composition of web services involves the web service selection/discovery and composition. This is achieved only by using the semantic web concepts.

Chapter 5

Problem Statement

5.1 Statement of the problem

Service Oriented Architecture (SOA) is the recent development in the field of computer science. This has made sure that the software is being developed efficiently and easily. Everything involves the online Internet activity. All the organizations have their websites and services. This helps to automate processes easily and also fetches global visibility. This Master's thesis will concentrate on the dynamic composition of these web services for any given requirement. In particular, an algorithm is developed to evaluate the dynamic composition. This technique is further checked for its quality of services being selected and their performance.

5.2 Purpose of the study

5.2.1 Overview

Service oriented architecture helps to integrate the web-based applications. It can also be mentioned as web services can implement a service oriented architecture. The advantage of web services is it provides the basic protocols which can be accessed by any system in the network. There are lots of web services which perform various functions are available free in the internet.

Web services are self contained, self describing modular applications that can be published, located and invoked anywhere in the network. Basically, this helps to develop the system quickly using the exiting services. As there are more web services available with the same functionality, it is important to choose the most appropriate one. Further a requirement may involve more than one service, the composition should be done properly.

5.2.2 Web service composition

Web service is the recent trend of the information technology industry. Web service composition is to compose different web services to achieve the given requirement. Web service composition helps in composing the complex web services quickly. Thus, it produces a powerful web service product.
Services can be composed both statically and dynamically. The service chosen during the design time is the static approach. Most of the current technology works in the static method. There are only few case studies and no practical experiment for dynamic implementation of web service composition. In dynamic composition, the services are selected during the run-time. The services are being search in the repository and then the best fit service is being selected.

5.3 Web Service Definition Language(WSDL)

Web service definition Language(WSDL) helps to provide a way to communicate with the service providers and the web server/client. There are certain specifications which are required for the WSDL. Limthanmaphon and Zhang has described about this in their paper “Web Service Composition with Case-Based Reasoning”. The authors mention that there are seven elements required for establishing a service in network. One of the most commonly used language is WSDL.

- Type provides information about the data type definitions to describe the message exchanged.
- Message has the details of the abstract definition of data that has to be transmitted or communicated.
- Operation produces the abstract description of the action that the service can support. This refers to the input message and output message.
- Port Type is the operation set by one or more endpoints.

- Binding mentions the concrete protocol and data format specification for the operations and messages.
- Port refers to the URL address of the web service listening.
- Service is the collection of related ports

5.4 Quality of service and their performance

Web services involve the network protocol. There are lots of possibility that will affect the quality and performance of the composition of service. Web services involve the standards of protocols. So the quality of services(QoS) is priority for the service provider. As the web services are composed dynamically composed, the quality of the services are likely to be affected. It should be made sure that the web services QoS requirement emphasize on both functional and non-functional. This improves the performance, reliability, integrity, accessibility, interoperability and security.

In the industry QoS is being given high importance. There are issues of poor performance and scalabity due to wrong selection of services. There are many methods to improve the quality and responsiveness of the services. This thesis work will be concentrating only on the web composition framework's quality of services. The unique feature of dynamic web service composition requires efficient testing and selection. These web services are being described in web service specific interface definition language known as WSDL (Web Service Discovery Language). The web services quality should be checked both qualitatively and qualitatively.

5.5 Contribution

From the above, it is obliviously seen that service composition has no definite model. Even if there are few approaches, they are all static and not dynamic. The key concern has now become the service selection and their composition for any given requirement.

This thesis report will be concentrating on the new method which will help the developers to compose the services during run-time. Evaluation of the created new software will become simple. Further, the performance and quality of the service is also measured.

An experiment will be conducted to show the web service composition and its effectiveness. In the experiment, three web services are taken into account, which together will be help to make the module working in a better way. This will enable the developer to write fewer lines of code for development. The application is being created in a simpler and quicker manner.

5.6 Industrial purpose

The introduction of this new idea in the field of service-oriented development, will allow developers to produce better software applications quickly and more efficiently. In the development side, the coders have to write few thousand line of code for development. Whereas if this is being used, the developers can use the existing web services with the functionality that would match the requirement for the development.

Even though this gives the design of the software application, it almost has completed the work except the database connectivity. The design of the software application can be shown to the client. When the client approves, the coder has to just work on the database repository. This not only reduces the time of development, the quality of the product would also be better.

Chapter 6

Proposed Method

6.1 Framework for the design procedure

The authors Muhammad Ahtisham Aslam, Jun Shen, Soren Auer, Michael Hermann have mentioned in their paper “An Integration Life Cycle for Semantic Web Services Composition”, about different approaches and also a life cycle model. The foundation of this framework model is being used in this thesis.

The life cycle of the semantic web services integration and composition is being given below. The life cyle is based on top down approach which means the business requirement analysis is at the start followed by the web service composition and ends at the execution. The multiple process that exists are the development of the business process, adding technical and business constraints to processes, producing the composition model and finally deploying the project and executing is the final stage.

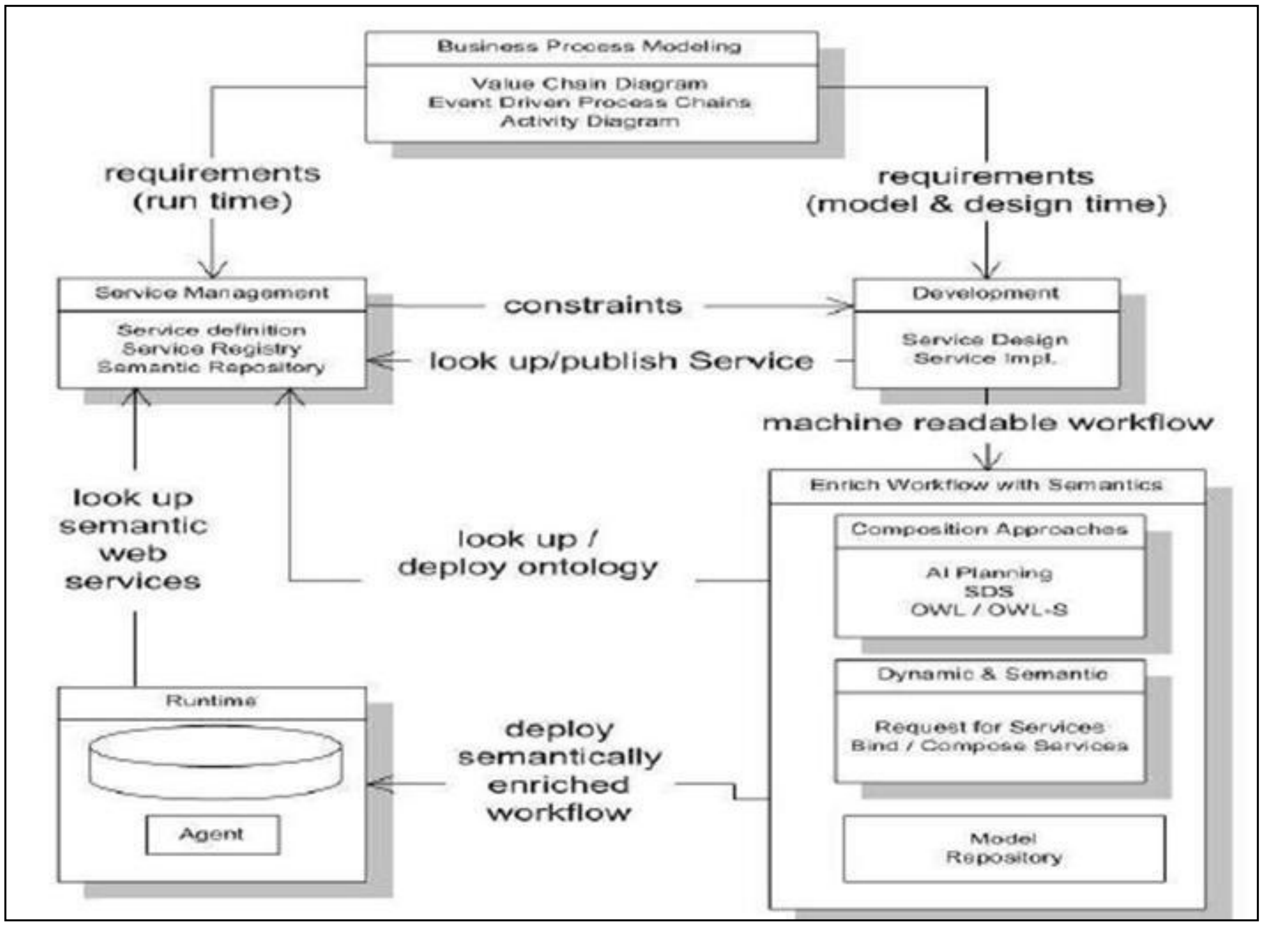


Figure 6.1: Framework Model of Web Service Integration [4]

6.2 Methodology

Any framework or architecture needs to have a design phase. Though, SOA does not have prescribed designing method or architecture for development, based on the idea obtained from the software process model and from [8], a similar method is proposed for designing the services. This process is iterative which helps in determine the user tasks and how to proceed further. The proposed method will have following steps and functions.

The framework model is being modified with the service selection and service composition.

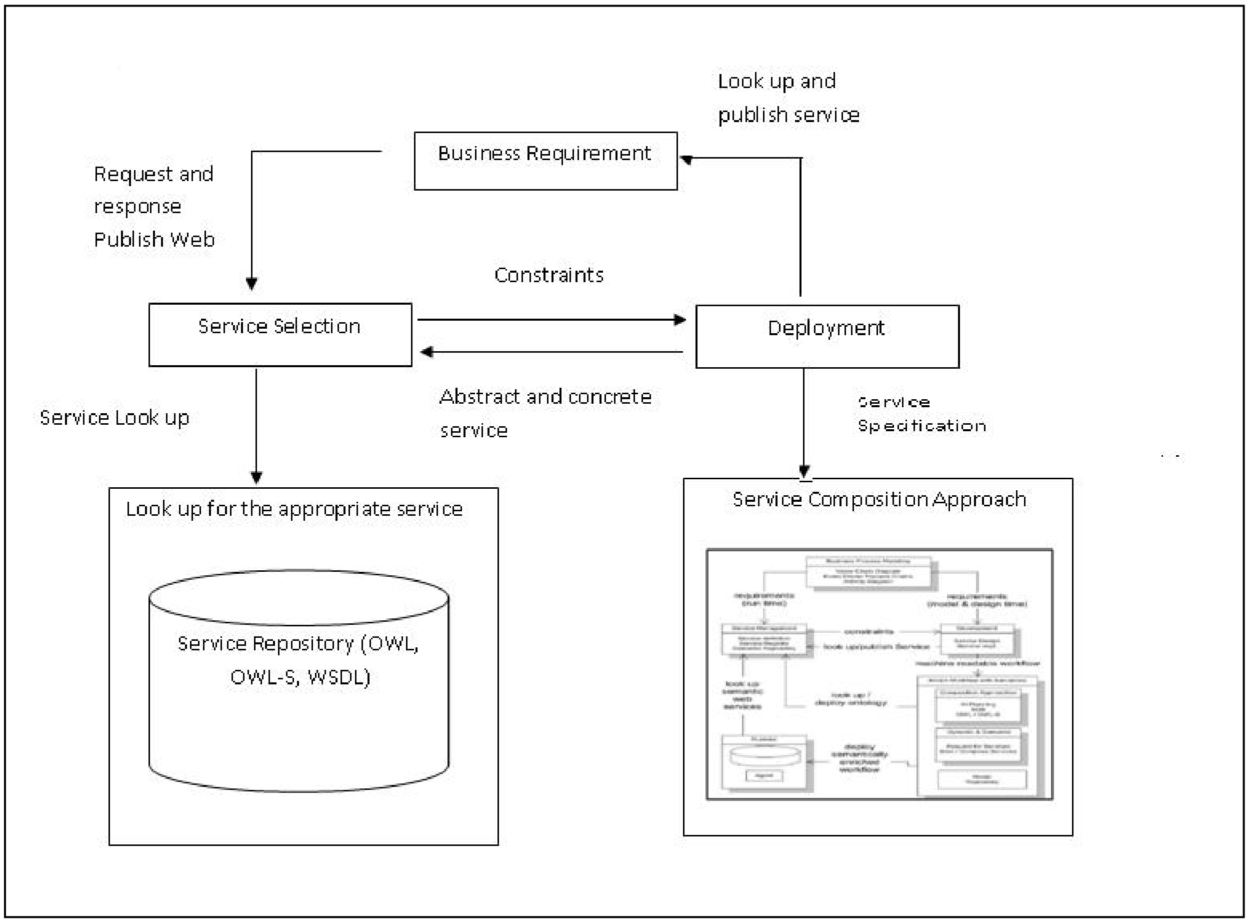


Figure 6.2: Framework of the proposed method

The flow chart explains the structure in which the data is being executed for the proposed method.

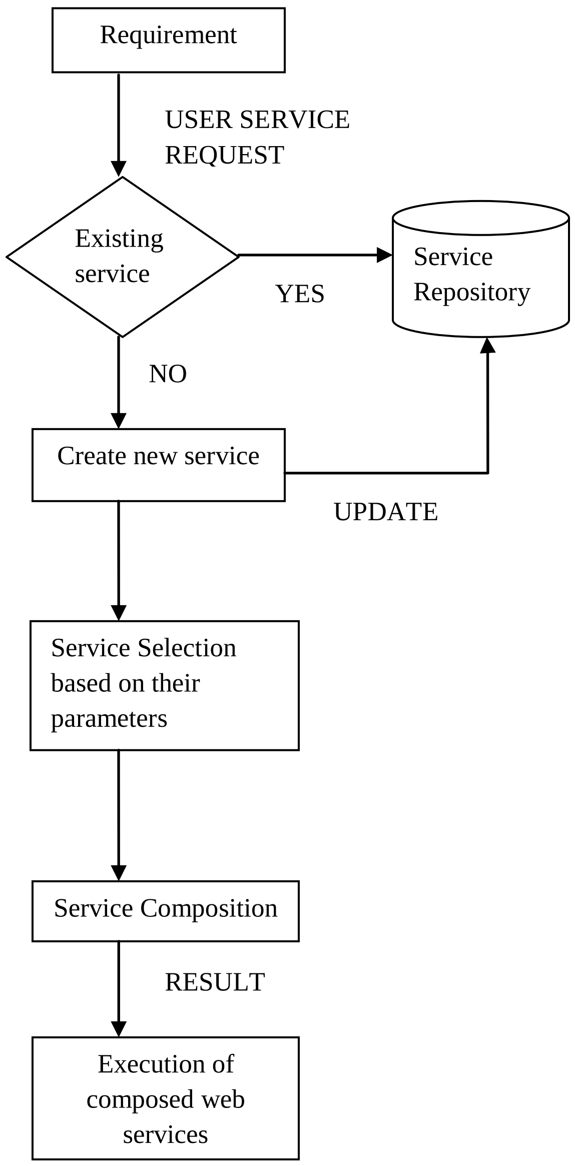


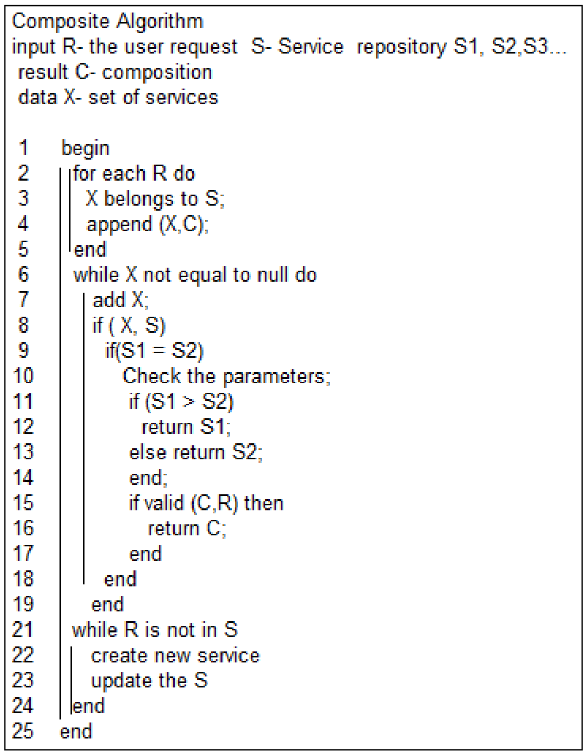
Figure 6.3: Flowchart of the proposed method

The following gives an outline of how the algorithm functions.

1. Analysis of the requirement
2. Discovers all the potential services for the requirement otherwise knows as the service selection which is based on the description of the services which are stored in the ontology repository
3. Best fit service is being composed for evaluation
4. Performance of the composition is being measured.

Advantage of this new approach is that incorporates both the work to make it a complete software system. Further the service composition based on human computer interaction is a new concept. Further evaluation of the services is done. The following expresses the process of algorithm in detail.

Advantage of this new approach is that incorporates both the work to make it a complete software system. Further the service composition based on human computer interaction is a new concept. Further evaluation of the services is done.



```
Composite Algorithm
input R- the user request  S- Service  repository S1, S2,S3...
 result C- composition
 data X- set of services

begin
 for each R do
   X belongs to S;
   append (X,C);
 end
 while X not equal to null do
   add X;
   if ( X, S)
     if(S1 = S2)
       Check the parameters;
       if (S1 > S2)
         return S1;
       else return S2;
       end;
       if valid (C,R) then
         return C;
       end
     end
   end
 while R is not in S
   create new service
   update the S
 end
end
```

Figure 6.4: Algorithm of the proposed method

Step 1 - Analysis of services

The available or existing services are all being analyzed. User will not have a chance to choose the service. The user will be giving the requirement to the developer for developing a new application. Or an application through which user can specify the requirement and system choosing the best fit service for the application.

Step 2 – Selection of services

As all the applications are based on internet usage, service composition will be using web services for developing an application or new product. According to W3C, web services are termed as software system interoperable between machines over a network. They have classified web services into two types.

(1) Simple web service: It provides a request type of functionality and does not support transactions.

(2) Complex web service: It provides the framework for business-to business collaborations and business process management.

Simple web service will be used in most of the cases as it is point-to point messaging and provides web security.

Step 3 – Composition of services

Service composition is based on selection of services. The composition of service can be automatic or semi-automatic or manual. There are different approaches followed. Some of the methods are discussed below. All these approaches will be using SOAP or some other protocol, RDF schema, WSDL, XML. Service structure cannot be understood without technical knowledge. The semantic knowledge of the services is necessary for the composition. This semantic knowledge is described using WSDL(Web service Description Language). This is then represented in XML for service transfer. The transfer takes place by the use of the SOAP protocol. Mostly, SOAP uses HTTP for transfer. Universal Description, Discovery and Integration is a registry which will now store the WSDL documents and SOAP messages which can be reused later.

Step 4 – Quality of services

After the selection of services, it is necessary to validate its flexibility, scalability, reliability and performance. This is altogether known as Quality of Services (QoS). This checks the network in which the web services are transferred. There are different algorithms to check the quality of the services.

Step 5 - Mapping with the model

After the service registration, service design and composition as a next step, the web service is supposed to get mapped to perform the action of building a new system. All the new web services which are identified are saved in the service repository for future purpose. This becomes very useful for creating a new system in an easy method. An overview of the web service architectural model is given below.

Chapter 7

Evaluation and Analysis of Experiment

7.1 Experiment Analysis

The web service composition is being determined by conducting an experiment to demonstrate the advantages of the dynamic web service composition. Through this experiment, the composition is being addresses and its issues. The SOA software model is the basic architecture for both the service providers and service requestors. Three web services are being used for the web service composition. All the three web services have different functions. These unique functionality services are being combined together to perform an operation together.

The first part of the experiment deals with the identifying appropriate services for the given requirement is being selected. The second part of the experiment demonstrates the composition of the services. This will make sure that the web services are being composed dynamically ie during the run time. The production of the software quality is better.

7.2 Experiment requirement

In order to successfully deploy this experiment involving the web services and its composition, it is a must that the following hardware and software requirements are supposed to be there. The following tables represent the hardware and software requirements respectively.

Product	Description	Specifications
Personal Computer	A computer with minimum configurations.	The exact specifications are it should have • 1 gigahertz or faster processor. • 1 gigabyte RAM or more. • 16 GB available hard disk space • DirectX 9 graphics device or higher (to view the output)

Table 7.1: Hardware Requirements

Software Requirements

Software Title	Description	Availability
Microsoft Windows Vista	Operating system in the computer	Available from Microsoft
NetBeans	This IDE helps in the development	http://www.netbeans.org
Java SDK	The java development language kit (latest version)	http://www.sun.com
GlassFish server	A server is needed to access the web services.	http://www.glassfish.java.net

Table 7.2: Software Requirements

7.3 Sample Services

This experiment requires 3 sample web services with related functionality to perform the web composition test. The experiment is completed by using these services that has been created. All these services have their own different functionality. The main objective is based on the product search results, cost payment and shipping.

Another service with the same functionality of product search is used. This is used so that the dynamic service composition is able to identify the most appropriate service for the composition. The implementation method includes these 3 services which is based on REST (Representational State Transfer). This attempts to describe the HTTP or protocols for constraining an interface. The architecture based on REST can also use WSDL to describe the basic architecture. The advantage of REST s that it helps the developers to use the GET,POST, PUT,DELETE operations to interact with other web services.

7.4 Case Study

The case study is on the ebusiness/online shopping which is popular these days. The requirement from the end user is being elicited using onotology. The developer using the application package which will contain the service client and its adapters. The appropriate service is being searched in the repository. The service which matches the requirement is being stored in the list of selected services. Finally when all the requirements' services are being collected, they are made together for the application package. The following class diagram is based on the ecommerce example. This explains the flow of the process in the application.

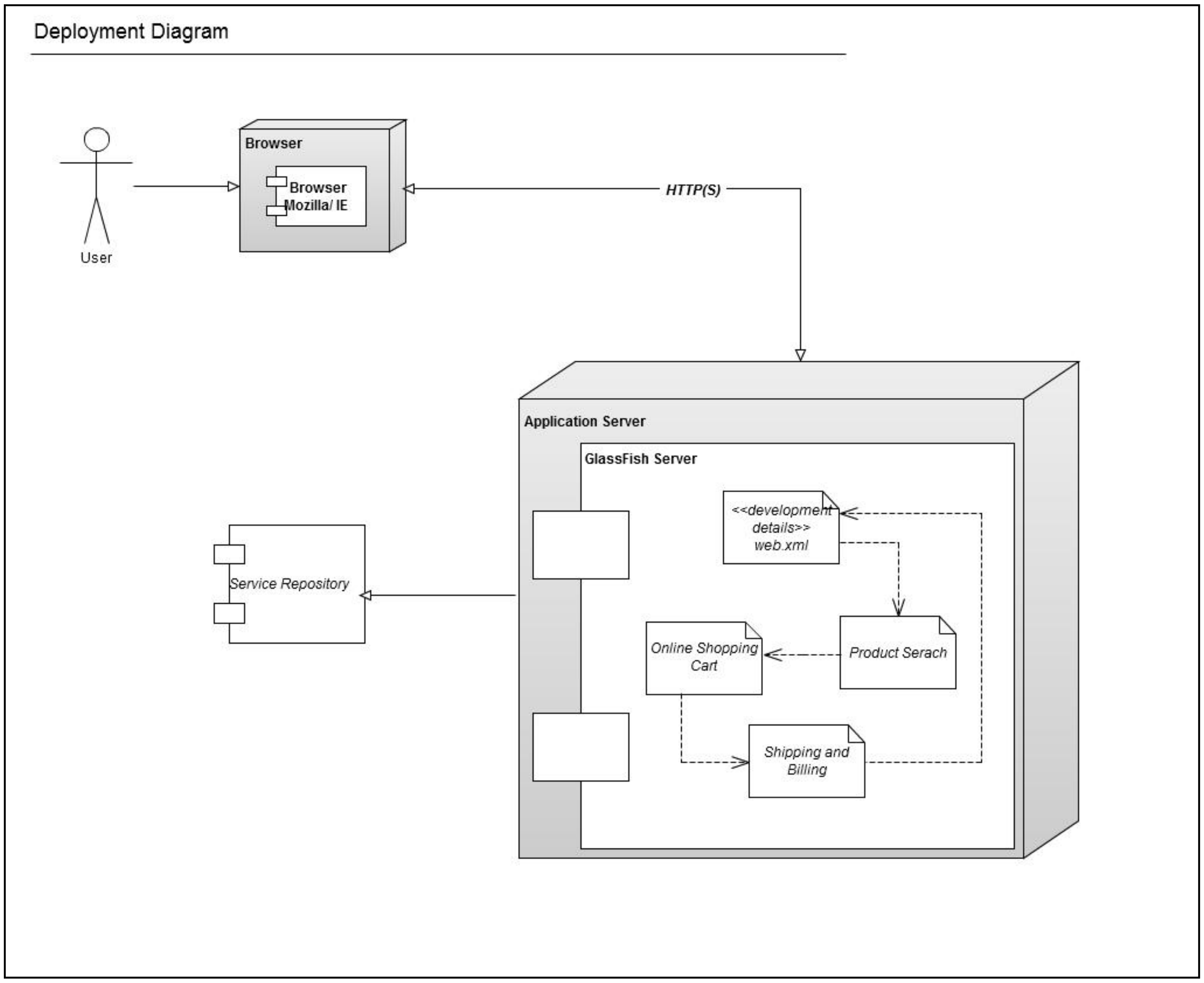


Figure 7.1: Deployment Diagram for the case study

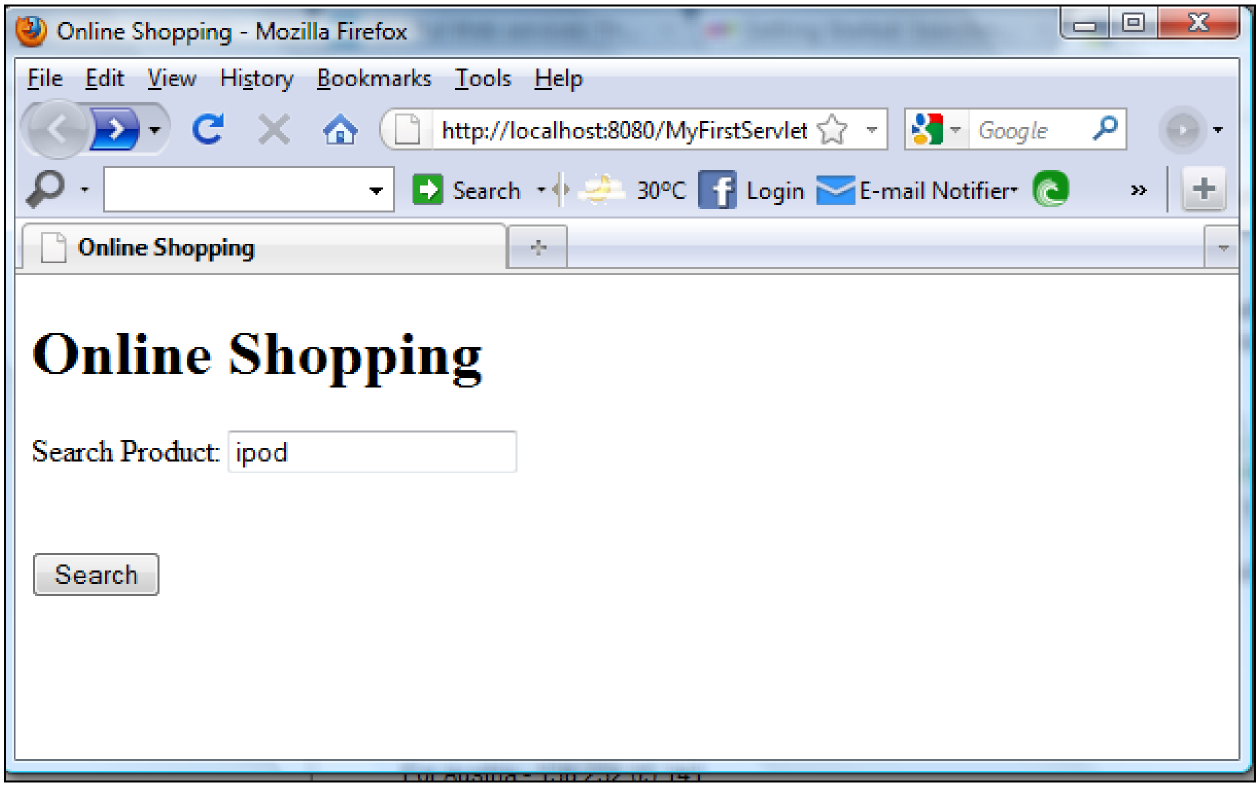
Online shopping is increasing these days. Web service composition experiment is conducted on this ecommerce. The basic shopping website will require three main services (1) search for the products (2) display the search products with their cost (3) billing and shipping the product. The main page is being developed. The search product web service is being invoked here. In the search product criteria, the product name is being typed. According to the product name, the search results would be displayed. During that time, the appropriate web service is being called. As the final step, a product is choosing for billing and shipping. This web service is being called. The end result of this experiment is that the web page is being designed user interactively easily by just using the available web services.

Suppose the repository has two kinds of product search, the service selection chooses the most appropriate service. For example if the client wants electronics product service and the service repository has two kinds of the service searches one for electronics goods and another for video games. The related service product search results service is being chosen. The following architecture structure will give a better idea about this concept.

7.5 Results

The proposed methodology is being tested on an online shopping system. As the initial step 3 services are being considered for the dynamic service composition. As a next step, another service was created with the same functionality of the credit card. The best or the most appropriate web service is being chosen.

The following figures show the result of each web service being triggered. The first page is a .jsp page which gives the basic outline of the page. In this case, a product is being search, so a search page is being displayed.


Online Shopping - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8080/MyFirstServlet
Google
Search
30°C
Login
E-mail Notifier
Online Shopping

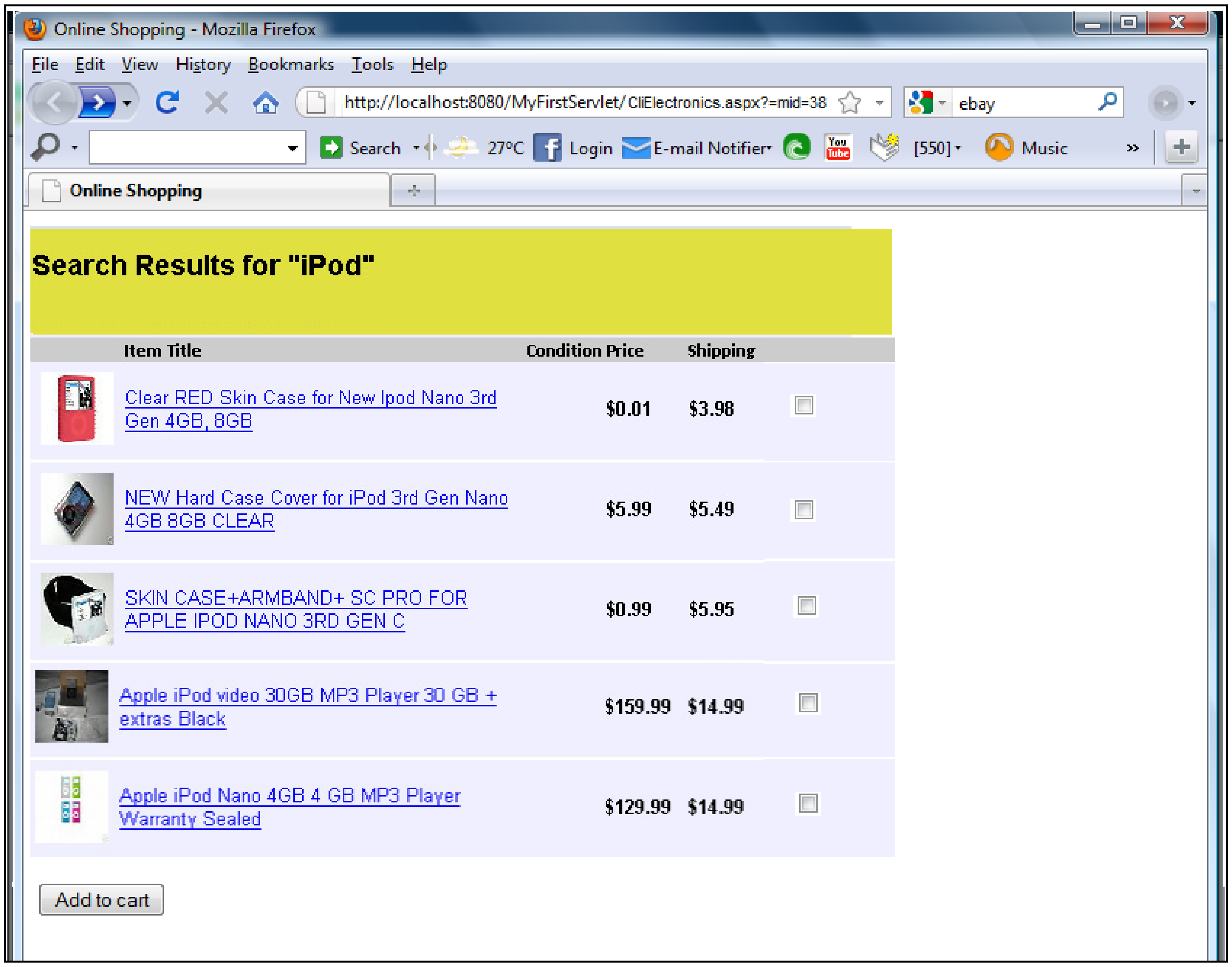
Online Shopping

Search Product: ipod

Search

Figure 7.2: User Interface home page

The product ‘ipod’ is searched. The search results (with their cost) is the following screenshot. The product search results web service is being accessed and results of the searched keyword is being provided. If the search product is not there, a screen displaying “the search product is not found” would be displayed.


Online Shopping - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8080/MyFirstServlet/CliElectronics.aspx?=mid=38
ebay
Online Shopping

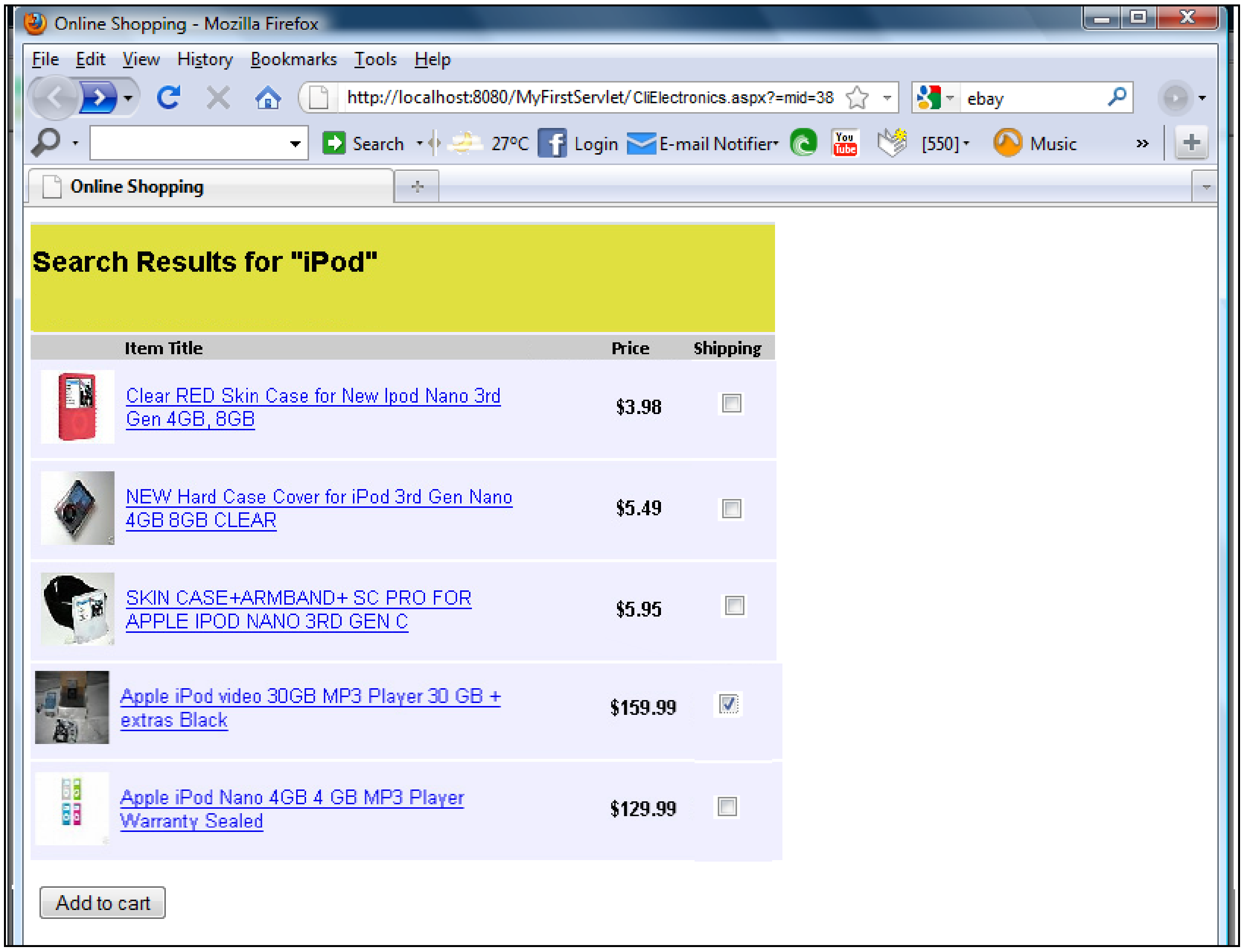
Search Results for "iPod"

	Item Title	Condition	Price	Shipping	
	Clear RED Skin Case for New Ipod Nano 3rd Gen 4GB, 8GB		$0.01	$3.98	☐
	NEW Hard Case Cover for iPod 3rd Gen Nano 4GB 8GB CLEAR		$5.99	$5.49	☐
	SKIN CASE+ARMBAND+ SC PRO FOR APPLE IPOD NANO 3RD GEN C		$0.99	$5.95	☐
	Apple iPod video 30GB MP3 Player 30 GB + extras Black		$159.99	$14.99	☐
	Apple iPod Nano 4GB 4 GB MP3 Player Warranty Sealed		$129.99	$14.99	☐

Add to cart

Figure 7.3: User Interface search page

As the next step, a product is being chosen that should be billed and shipped. The screenshot below shows the checked product.

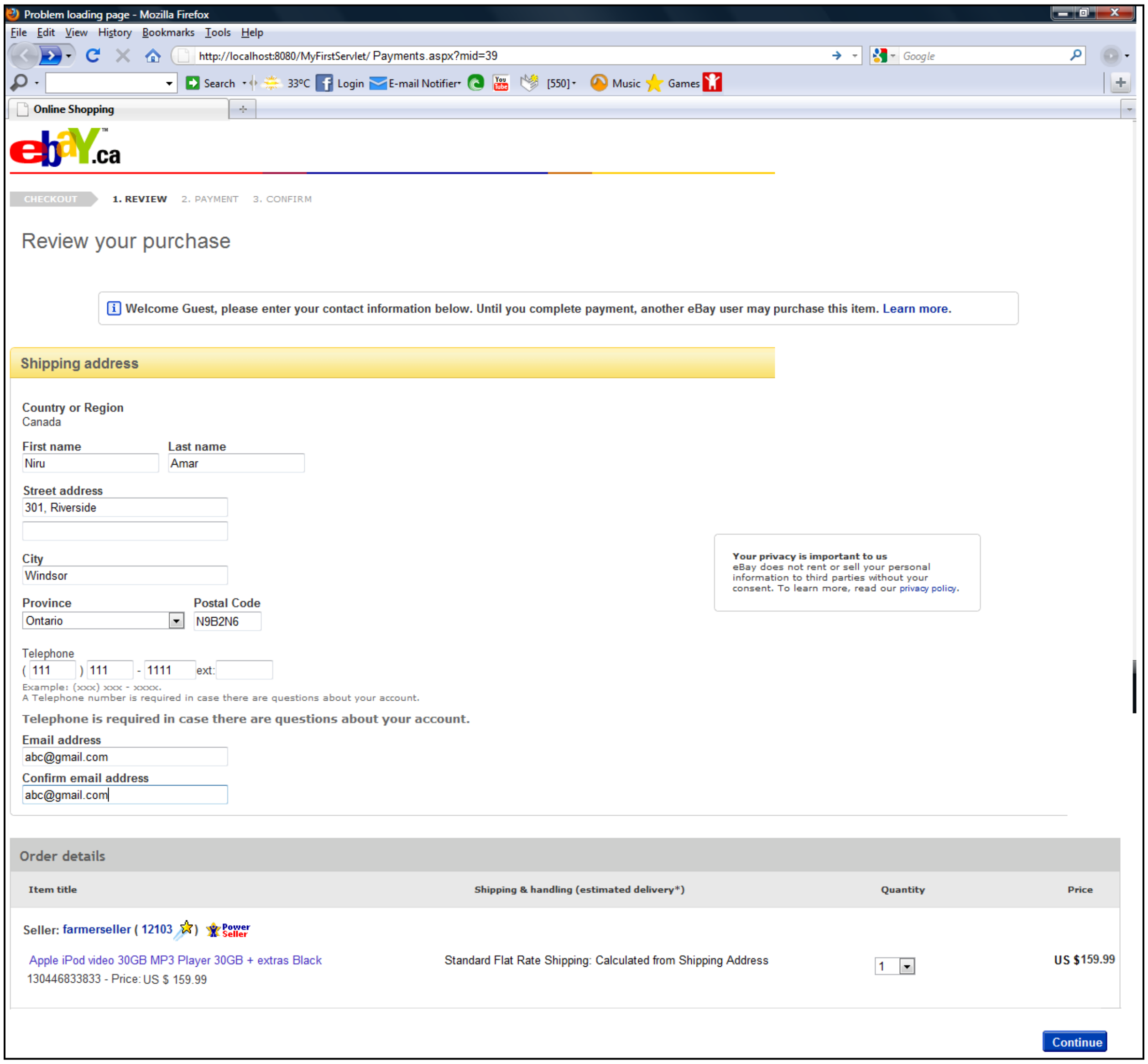

Online Shopping - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8080/MyFirstServlet/CliElectronics.aspx?=mid=38
ebay
Search 27°C Login E-mail Notifier [550] Music
Online Shopping
Add to cart

Search Results for "iPod"

Item Title	Price	Shipping
Clear RED Skin Case for New Ipod Nano 3rd Gen 4GB, 8GB	$3.98	☐
NEW Hard Case Cover for iPod 3rd Gen Nano 4GB 8GB CLEAR	$5.49	☐
SKIN CASE+ARMBAND+ SC PRO FOR APPLE IPOD NANO 3RD GEN C	$5.95	☐
Apple iPod video 30GB MP3 Player 30 GB + extras Black	$159.99	✓
Apple iPod Nano 4GB 4 GB MP3 Player Warranty Sealed	$129.99	☐

Figure 7.4: User Interface search page 2

Finally step includes the billing address and the credit card details. When the “add to cart” button is clicked, it invokes the third web service credit card details.


Problem loading page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8080/MyFirstServlet/ Payments.aspx?mid=39
Online Shopping
ebay.ca
CHECKOUT 1. REVIEW 2. PAYMENT 3. CONFIRM
Continue

Review your purchase

Welcome Guest, please enter your contact information below. Until you complete payment, another eBay user may purchase this item. Learn more.

Shipping address

Country or Region
Canada

First name	Niru
Last name	Amar
Street address	301, Riverside
City	Windsor
Province	Ontario
Postal Code	N9B2N6
Telephone	(111) 111 - 1111 ext:
Email address	abc@gmail.com
Confirm email address	abc@gmail.com

Example: (xxx) xxx - xxxx.
A Telephone number is required in case there are questions about your account.

Telephone is required in case there are questions about your account.

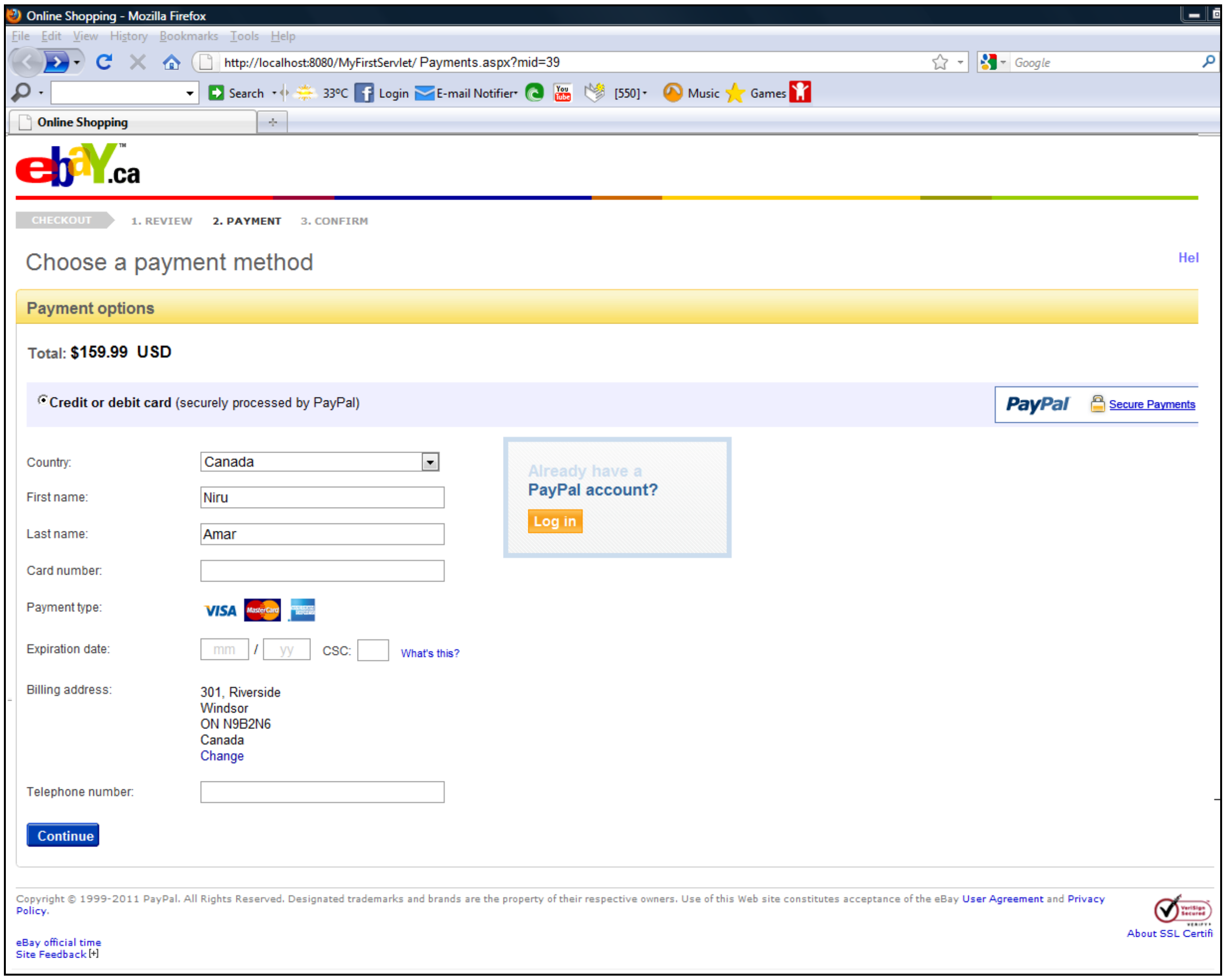
Your privacy is important to us
eBay does not rent or sell your personal information to third parties without your consent. To learn more, read our privacy policy.

Order details

Item title	Shipping & handling (estimated delivery*)	Quantity	Price
Seller: farmerseller (12103) Power Seller			
Apple iPod video 30GB MP3 Player 30GB + extras Black 130446833833 - Price: US $ 159.99	Standard Flat Rate Shipping: Calculated from Shipping Address	1	US $159.99

Figure 7.5: User Interface payment options

The payment of the product being purchased is being made.


Online Shopping - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8080/MyFirstServlet/ Payments.aspx?mid=39
Online Shopping
CHECKOUT 1. REVIEW 2. PAYMENT 3. CONFIRM

Choose a payment method

Payment options

Total: $159.99 USD

Credit or debit card (securely processed by PayPal)

PayPal Secure Payments

Already have a PayPal account? Log in

Country:	Canada
First name:	Niru
Last name:	Amar
Card number:	
Payment type:	VISA MasterCard American Express
Expiration date:	mm / yy CSC: What's this?
Billing address:	301, Riverside Windsor ON N9B2N6 Canada Change
Telephone number:	

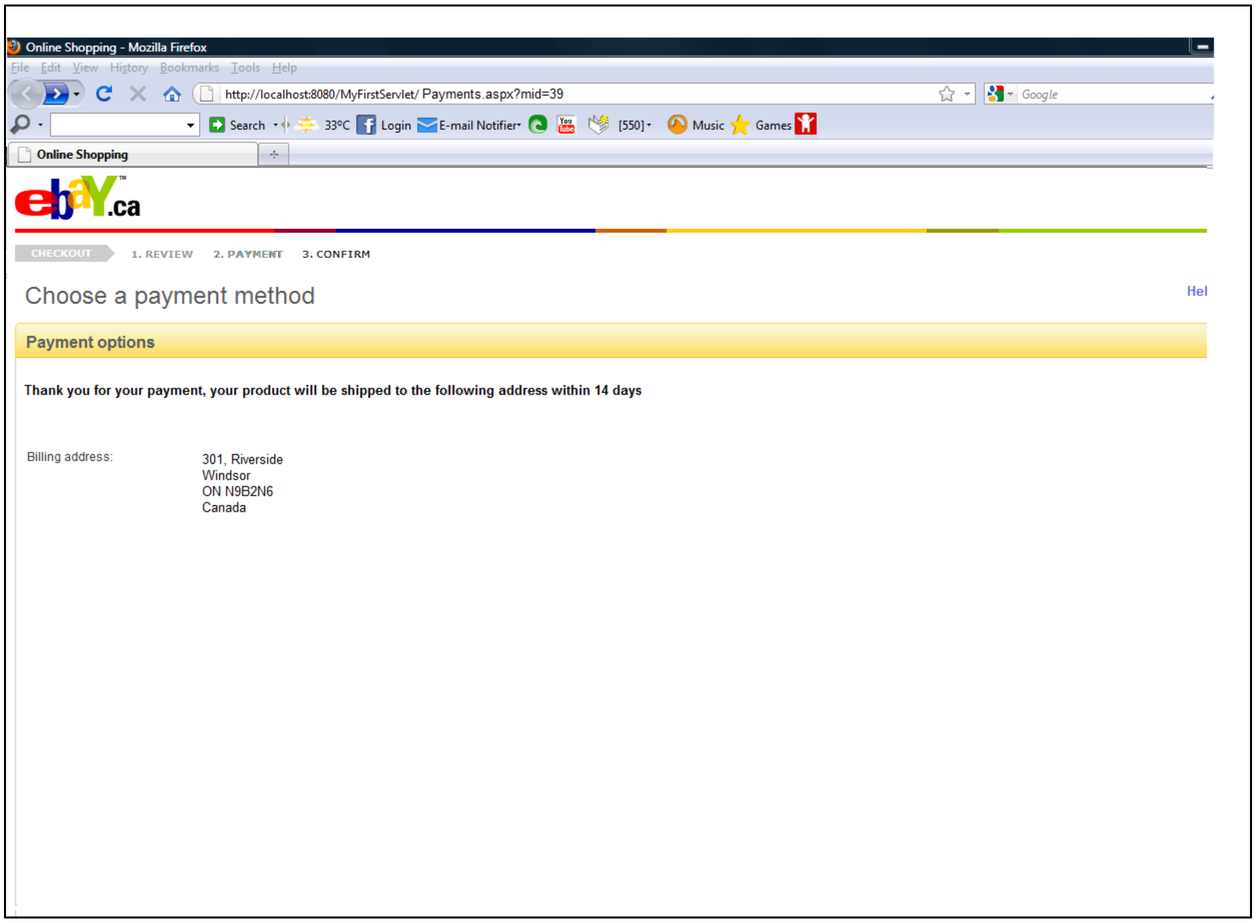
Continue

Copyright © 1999-2011 PayPal. All Rights Reserved. Designated trademarks and brands are the property of their respective owners. Use of this Web site constitutes acceptance of the eBay User Agreement and Privacy Policy.

eBay official time
Site Feedback

Figure 7.6: User Interface payment options2

Finally, the confirmation says that the payment is made successful and the product will be delivered.


Online Shopping - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8080/MyFirstServlet/ Payments.aspx?mid=39
Online Shopping
ebaY.ca
CHECKOUT
1. REVIEW
2. PAYMENT
3. CONFIRM
Hel

Choose a payment method

Payment options

Thank you for your payment, your product will be shipped to the following address within 14 days

Billing address: 301, Riverside
Windsor
ON N9B2N6
Canada

Figure 7.7: User Interface confirmation page

This is a design of the actual software system. If these services are being connected to a software system it makes the system complete.

7.6 Analysis

The service selection takes place at the final step of the implementation of the case study taken. The best service is being chosen. For the given case study a user gives the credit card details for the billing. At this point the appropriate credit card service should be triggered.

This process takes place in the payment gateway. A payment gateway is an e-commerce application service provider that authorizes payments. Also this makes sure that the credit card details are secure. Users have the choice of selecting the services. Some users may be willing to pay more cost for a better service but others may prefer to sacrifice performance at reduced cost. The client is being enquired with these factors for the ideal selection of service. According to the case study taken, all these details are being provided before itself and the two web services for credit card payment has their own performance difference. If the client is choosing “Master Card” web service i.e. service A, the performance is shown to be better than the service B i.e. “VISA”

The performance of the service sensitivity is done here. The following table gives the cost and performance value.

	Service A	**Service B**
Cost	7	3.50
Avg Response Time	350	700
Avg Data Transferred	10	20
Performance Value with each category weighted at 33%	82.5%	66%

Table 7.3: Performance table

Chapter 8

Conclusion and Future Work

Service Oriented Technology (SOA) is an emerging technology that has been adapted and embraced by businesses and developers everywhere. It is said to be the future of the World Wide Web. SOA strengthens reusability through its use of loose coupling and services. The future will be mostly on web based and SOA will be playing a vital role. The process of developing new systems will reduce and the usage of existing services/techniques to build a system will come soon as it is easier because of less time consumption and with less of errors. Though there are some disadvantages of SOA, they overshadow when compared with the advantages.

Web service composition and design can be done automatically using different algorithms. Their performance and quality of service can also be measured. Thus, a reliable and efficient system can be developed. Further the introduction of service sensitivity and dynamic service composition will help to develop a better software system which is the present day trend. This can be further lead for software visualization

References citied

1. Thomas Weise, Steffen Bluel, Diana Comes, Kurt geihs, "*Different approaches to Semantic Web Service Composition*", proceedings in IEEE Third International Conference on Internet and Web Applications and Services,2008.

2. Carlos Parra, Anthony Cleve, Xavier Blane, Laurence Duchien, "*Feature-Based Composition of Software Architecture*", proceedings in Springer-Verlag Berlin Heidelberg 2010, pp 230-245.

3. Haiyun Sun, Xiaodong Wang, Bin Zhou and Peng Zou, "*Research and Implementation of Dynamic Web Services Composition*", proceedings in Springer-Verlag Berlin Heidelberg 2003, pp 457-466.

4. Muhammad Ahtisham Aslam, Jun Shen,Soren Auer, Michael Hermann, "*An Integration life Cycle for Semantic Web Services Composition*", proceedings of the 2007 11th International Conference on Computer Supported Cooperative Work in Design, pp 490-495

5. E.Sirin,B.Parsia, and J.Hendler, "*Temple-based Composition of Semantic Web Service*", proceedings in AAAI Fall Symposium on Agents and Semantic Web, Virginia,USA, November 2005

6. Anne Martens, Heiko Kozioleky, Steffen Beckerz, Ralf Reussnerz , "*Automatically Improve Software Architecture Models for Performance, Reliability, and Cost Using Evolutionary Algorithms"* proceedings in ACM conference January 28–30, 2010, San Jose, California, USA.

7. Thomas Fischer, Fedor Bakalov, Andreas Nauerz, Martin Welsch , *"An Evolutionary Algorithm for Automatic Composition of Information-gathering Web Services in Mashups",* proceeding in 2009 Seventh IEEE European Conference on Web-services, pp 39-48.

8. Gexin Li, Shuiguang Deng, Haijiang Xia, Chuan Lin , *"Automatic Service Composition Based on Process Ontology"* , proceeding in Third International Conference on Next Generation Web Services Practice, pp 3-6.

9. Shuiguang Deng, :Zhaohui Wu, Ying Li, *"ASCEND: A Framework for Automatic Service Composition and Execution in Dynamic Environment",* proceedings in 2004 IEEE International Conference on Systems, Man and Cybernetics, pp 3457-3461.

10. L.Richardson and S.Ruby, RESTful web services. O'Reilly Media,Inc, 2007

11. E. Michael Maximillien, Munindar P.Singh
http://www.csc.ncsu.edu/faculty/mpsingh/papers/mas/aamas-socabe-05.pdf

12. Mazen Malek Shiaa , Jan Ove Fladmark , Benoit Thiell, *"An incremental graph-based approach to Automatic Service Composition",* proceedings in 2008 IEEE International Conference on Services Computing, pp 394-404.

13. Philippe Kruchten University of British Columbia, *"An Ontology of Architectural Design Decisions in Software-Intensive Systems".*

14. Sedighe Moosavi, Mir Ali Seyyedi, Nasrollah Moghadam *,"A Method for Service Oriented Design"*, proceedings in 2009 Sixth International Conference on Information Technology: New Generations, pp 290-295.

15. Jinghai Rao and Xiaomeng Su , *" A Survey of Automated Web Service Composition Methods".*

16. C. Y. Knaus, *"Feature - interaction design for software engineering: Boost into programming future,"* Interactions, vol. 15, no. 4, pp. 71-74, July 2008.

17. C. Lai and W. Liou, *"A service-oriented architecture for constructing ontology-based learning objects repository,"* in Proceedings of the Ninth IEEE international Symposium on Multimedia Workshops, 2007, pp. 349-355.

18. W. Tsai, Q. Huang, J. Xu, Y. Chen, and R. A. Paul, *"Ontology-based dynamic process collaboration in service-oriented architecture,"* in Proceedings of the IEEE international Conference on Service-Oriented Computing and Applications, 2007, pp. 39-46.

19. Michael N.Huhns ,Munindar P. Singh, *"Service-Oriented Computing: Key Concepts and Principles"*, in proceedings of IEEE Internet Computing Journal January-February 2005, pp 75-81.

20. Michael Maximillien, Munindar P.Singh *"A Framework and Ontology for Dynamic Web Service Selection",*in proceedings of IEEE Computer Society September-October 2004, pp 84-93.

21. Hongyu Sun, Robyn R.Luz, Samik Basu "*Product-Line-Based Requirements Customization for Web Service Compositions "* in proceedings Journal of Systems and Software June 2008, pp 855-867.

22. L. Breierova and M. Choudhari. "*An introduction to sensitivity analysis*" . Prepared for the MIT System Dynamics in Education Project, 1996

23. D.J. Pannell. Sensitivity analysis: strategies, methods, concepts, examples.

24. A. Saltelli, K. Chan, E.M. Scott, et al. Sensitivity analysis. Wiley New York, 2004.

Vita Auctoris

Niruppama Amarnath was born in the year 1985 in Tamil Nadu, India. She received her Master' of Sciences with the specialization on Software Engineering from VIT University, Tamil Nadu, India. She joined the University of Windsor in January 2010 and is currently a candidate for Master's degree in Computer Science under the supervision of Dr.Yuan.