

Applying SOA to an Ecommerce system

Hanafi Mughrabi

Kongens Lyngby
IMM-MSc-2007

IMM

Abstract

Surveys indicate that small businesses are not adopting SOA in the same extent as large enterprises. However there are only few studies examining how small businesses planning to adopt SOA can do this best.

This report examines different approach of service development, in order to evaluate and recommend the most suitable in the context of small ecommerce businesses migrating to SOA, by an incremental migration strategy.

The report describes the background of SOA, small business ecommerce and examines literature in regards to service development approaches for SOA migration. Important aspects in regards to small ecommerce businesses are identified in order to evaluate a case implementation.

The case implementation deals with the important ecommerce functionality of searching a product catalog. Two different services are developed and evaluated based on two service development approaches identified to be feasible.

The thesis concludes with recommending a redevelopment approach, based on its high degree of extendibility and low dependency of the non-SOA ecommerce system, from which the service functionality was extracted.

Keywords: Service development approach, migration, SOA, Web services, small to medium businesses, small ecommerce business.

Preface

This thesis was prepared at Informatics Mathematical Modelling, the Technical University of Denmark in partial fulfilment of the requirements for acquiring the MSc degree in engineering, in the period from April 16, 2007 to September 27, 2007.

The report documents the master's thesis: Applying SOA to an ecommerce system. The report primarily centre it attention on identifying, and evaluating service development approaches, in order to find the most suited for small ecommerce businesses, planning to incrementally migrate to SOA.

All source code for the two projects can be found at the following address:

http://www.student.dtu.dk/~s001357/Thesis_Code

Acknowledgement

I would first like to thank Informatics Mathematical Modelling, at the Technical University of Denmark, who made this thesis possible. I would also like to thank my supervisor Niels Ole Christensen, for his help, guidance and patience. His criticisms and contributions were concrete and helped in the formulation and completion of this thesis.

Table of content

Abstract	iii
Preface	v
Acknowledgement	vii
1 Introduction	1
1.1 Introduction	1
1.2 Motivation	2
1.3 Problem statement.....	2
1.4 Limitations/Scope.....	3
1.5 Methodology	3
1.6 Contribution	4
1.7 Definitions of terms and concepts	4
1.8 Readers guide.....	5
1.9 Structure of the report	5
2 Background.....	7
2.1 Service Orientation paradigm.....	7
2.1.1 SOA defined.....	7
2.1.2 SOA principles	7
2.1.3 SOA participants	9
2.1.4 Web services as SOA enablers	9
2.1.5 Why SOA	11
2.2 Migration	11
2.2.1 SOA migration.....	12
3 Small business ecommerce	13
3.1 Small businesses:	13
3.2 Adoption of ecommerce	13
3.3 Importance of exposure	14
3.4 Current small ecommerce business:	14
3.5 Important aspects in migrating to SOA.....	15
4 Literature review	17
4.1 Scope of the review	17
4.2 Keeping existing app. infrastructure	17
4.3 Integrating legacy systems in services	18
4.4 Summary	19
5 SDAs for SOA migration	21
5.1 Wrapping approach.....	21
5.2 Redevelopment approach	21
5.3 Hybrid approach.....	22
5.4 Migration approach evaluation criteria.....	22
6 Application of approaches.....	25
6.1 Purpose	25
6.2 Limits	25
6.3 Requirements analysis.....	26
6.3.1 Business goal & stakeholders	26
6.3.2 Open source ecommerce	26
6.3.3 Use case	26
6.4 Requirements specification	28
6.4.1 System	28
6.4.2 Approach.....	28
6.4.3 Service	28

6.4.4 Software.....	28
6.5 Design	29
6.5.1 Overall design	29
6.5.2 OSCommerce product search functionality.....	30
6.5.3 Service design.....	30
6.5.4 Contract	32
6.5.5 Communication and protocol.....	33
7 Implementation	37
7.1 OSCommerce implementation	37
7.1.1 OSCommerce structure.....	37
7.1.2 JSAS	38
7.2 Hybrid service implementation	40
7.2.1 NuSOAP library.....	40
7.2.2 Code excerpt.....	40
7.2.3 WSDL.....	42
7.3 Redevelopment service implementation.....	44
7.3.1 Tomcat Web server	44
7.3.2 Apache Axis	44
7.3.3 Code excerpt.....	44
7.3.4 WSDL.....	47
8 Evaluation of case implementation.....	49
8.1 Overall development complexity	49
8.2 System source code understanding	50
8.3 Ease of extendibility	51
8.4 Dependencies.....	52
8.5 Effort to achieve service principles	52
9 Discussion	55
9.1 Results.....	55
9.2 Usage of the service	56
9.3 Is having a good approach enough	56
9.4 Recommendation.....	56
10 Conclusion.....	57
Bibliography.....	59
Appendix A - Setup.....	62
Appendix B - Product search functionality.....	66
Appendix C – Service implementation	76
Appendix D - Test client implementation.....	88
Appendix E - Service test.....	92

List of Figures

Figure 1: SOA participants.....	9
Figure 2: Web service model	10
Figure 3: Online shopping flow.....	15
Figure 4: Solution Designs	29
Figure 5: Single result.....	31
Figure 6: Multiple results.....	31
Figure 7: Hybrid service.....	31
Figure 8: Redevelopment service	32
Figure 9: SOAP communication.....	34
Figure 10: OSCommerce shop	37
Figure 11: OSCommerce catalog folder.....	38
Figure 12: phpMyAdmin showing oscdb.....	39
Figure 13: TC1 output.....	93
Figure 14: TC2 output.....	93

List of Listings

Listing 1: WSDL format.....	33
Listing 2: SOAP message structure	34
Listing 3: SOAP request example	35
Listing 4: SOAP response example.....	35
Listing 5: Registering service with NuSOAP	41
Listing 6: prodSearch defined	41
Listing 7: Select part of sql statemet	41
Listing 8: Complete sql statement	42
Listing 9: fetching rows from appropriate oscdb tables.....	42
Listing 10: WSDL document for Hybrid service	43
Listing 11: Class productSearch	45
Listing 12: Definition of variables	45
Listing 13: Fetch data from oscdb	46
Listing 14: Return result.....	46
Listing 15: WSDL for Redevelopment service	47

List of Tables

Table 1: Evaluation criteria	24
Table 2: Test case	92
Table 3: Test case results.....	94

List of Acronyms

EDI:	Electronic Data Interchange
HTTP:	Hypertext Transfer Protocol
HTTPS:	Hypertext Transfer Protocol Secure
IT:	Information Technology
JSAS:	Joomla Stand Alone Server
PHP:	PHP Hypertext Preprocessor
SDA:	Service Development Approach
SMB:	Small and Medium Business
SOA:	Service-Oriented Architecture
SOAP:	Simple Object Access Protocol
UDDI:	Universal Description Discovery and Integration
WAMP:	Windows, Apache, MySQL, PHP
WSDL:	Web Services Description Language
WWW:	World Wide Web
XML:	Extensible Markup Language

Chapter 1

1 Introduction

1.1 Introduction

Traditionally, IT vendors have focused their attention on the large enterprise market with less attention towards the needs of the small and medium business (SMB) market [1].

This is caused by the lack of IT spending available pr. SMB, nevertheless the IT industry have successfully pushed IT adoption on SMBs accounting for 49 % of all IT spending¹. This may to some extent be related to the fact that in 2005 the SMB market accounted 70 % of the total worldwide employment, with emerging market players India and China reaching 72% - 73% of employment in the SMB market [1].

As the SMB IT market is still growing it is becoming a high priority market for vendors, who are becoming more aware of the SMB market position being just as important as the large enterprise market.

Predictions point to the SMB IT market segment to grow 8.5% in 2007 exceeding even the IT market growth rate itself and outpacing large enterprises [2].

Furthermore [2] predicts Software as a Service and service-oriented architecture (SOA) will play a crucial role in the SMB IT market segment, with the ability for better integration, increased flexibility, and cost reduction in development by reuse of existing services.

Even though this is the case SMBs don't have the same incentive as large organizations to change to SOA, as resources are much more limited.

In fact [3] shows that only a small amount of the SMB segment, have adopted SOA, but that there is a relative interest in researching the area.

¹ IDC Source, all data 2005 [3]

This could be argued as SMBs playing the waiting game, researching and monitoring the SOA evolution, waiting for the “right” time to enter the SOA stage.

1.2 Motivation

SOA is a promising architectural paradigm based on loosely coupled, self-contained software components (services), that can execute as independent entities, and cooperate together with other services to form complex business process. SOA claims to enable increased flexibility, interoperability, easy of integration and cost reduction in development.

Still, survey [1] shows that with only 19 % of the organizations having SOA in place, 11% implementing, 22 % still researching the technology and no activity of 47 % of the survey participants, there is still a long way from an ideal adoption of SOA...

For SOA to get a good foothold, the growing SMB IT segment must be included, and convinced that adoption of SOA is straightforward. It is therefore imperative to present SMBs wishing to adopt SOA, with an approach of doing this.

1.3 Problem statement

This thesis will centre its attention on investigating and comparing different service development approaches (SDAs), in order to find the most suitable in respects to incremental SOA migration for small ecommerce businesses, through services integration.

The problem statement for this thesis is therefore stated as follows:

Which service development approach is best suited for small ecommerce businesses employing an incremental SOA migrating strategy?

The value of examining this question, lies in giving small ecommerce businesses a clear SDA as part of an incentive, to encourage more small ecommerce

businesses to migrate to SOA, so they might benefit from the SOA paradigm quality attributes of reuse, interoperability, ease of development and integration.

1.4 Limitations/Scope

The scope of the thesis is limited to examining development approaches for service integration in order to determine which one will suit small business ecommerce best, if they follow an incremental SOA migration strategy.

Further the thesis limits itself to a simple case implementation, where a service for each identified approach will be implemented if it is seen fit for further investigation. Here the emphasis will be on evaluating the service development approaches, and not the architecture of the system as such. Therefore UDDI will not be implemented, though touched upon briefly in the background chapter.

The thesis further limits itself to the overall understanding of small ecommerce businesses, and will not go into details on the infrastructure, employment and/or financial aspects.

1.5 Methodology

The methodology used in this thesis is Qualitative research in the form of explorative case study, which best fits the study at hand, because it is most suited dealing with hypothetical issues, as findings in the subject area consists of opinions, lacking the statistical significance required in quantitative studies, as it is difficult to find statistical data about SOA migration and use in small businesses, as this is still fairly new area in this market segment.

Data to be analyses is mainly collected from published articles and reports, but magazines and newspapers engaged in this area, are also used to the extent that they present relevant information dealing with SDAs, SMBs, SOA or SOA migration.

These sources are mainly acquired by an extensive literature search conducted on several publication databases and on the internet, to gather as much relevant material as possible.

Subsequently an experimental method will be taken in form of a case implementation, designing, implementing, testing and evaluating the identified service development approaches. Results from this experimental method will be

the basis for recommending the approach most suited when small businesses ecommerce following an incremental SOA migration strategy.

1.6 Contribution

The thesis is meant to contribute with research on most suited SDAs for small business ecommerce migration to SOA when using an incremental strategy.

Further the thesis will deal with a simple case implementation of a simple but highly important functionality of ecommerce, giving a clear and practical insight of how service development to achieve SOA may be accomplished.

1.7 Definitions of terms and concepts

Migration:

Digital Libraries, by William Arms, (c) 2000 M.I.T. Press defines migration to be "Preservation of digital content, where the underlying information is retained but older formats and internal structures are replaced by newer".

The report will use Migration in the term "Service-Oriented Migration", as a way of migrating from a non-service oriented system to a system where partially or full system functionality can be exposed as services in a SOA.

Adoption:

Merriam-Webster [4] defines adoption as "to accept formally and put into effect".

In this report adoption is used in the term "SOA adoption" to describe the adoption of the SOA paradigm, to the extent, that companies have put SOA into effect to some extent.

Small and medium businesses:

The EU has started to standardize the concept. Its current definition categorizes companies with fewer than 50 employees or turnover under £10 million as "small", and those with fewer than 250 or turnover under £50 million as "medium". The thesis follows these standards.

Small business ecommerce:

Ecommerce performed by small businesses.

Small ecommerce businesses:

Small businesses, dealing in the ecommerce market.

Legacy Systems:

A legacy system is an antiquated computer system or application program which continues to be used because the user, typically an organization, does not want to replace or redesign it [5].

System Integration in Services:

Integrating of a given systems functionality into services. This is usually done with legacy systems, but can also be applied to more up to date systems.

1.8 Readers guide

- First time an abbreviation is introduced it will be written out full. This does not go for Service Oriented Architecture, which appear interchangeably as Service Oriented Architecture or SOA throughout the thesis.
- The report should be read sequentially to get the full understanding, but if one is only interested in the case implementation, chapters 2, 3 and 4 can be skipped.
- Code will be written in `currier new 10` to make it stand out from the rest of the thesis text.
- Text in *cursive*, is used when emphasizing words or phrases.

1.9 Structure of the report

Chapter 1 presented the area of research and research question. Scope and limitation, research method is presented next, followed by a description of the contribution the thesis will ad to the subject and definition of important terms & concepts.

Chapter 2 introduces the SO paradigm, its principles, and supporting technologies for realisation. This is followed by an introduction to migration in general and SOA migration.

Chapter 3 explores small business ecommerce, and which issues/aspects that have importance for small businesses ecommerce migrating to SOA.

Chapter 4 reviews related research, presenting the major ideas of SDA for SOA migration. The chapter concludes with a summary of research review results, setting the stage for further investigation.

Chapter 5 presents the different SDAs identified in the literature review chapter. This is followed by the definition of important assessment criteria for SDA in respect to small ecommerce businesses migrate to SOA.

Chapter 6 explores an case implementation where the identified approaches will be applied to services in an open source ecommerce system. Requirements and design are presented and described in this chapter.

Chapter 7 deals with the actual implementation of the services and presents the most important implementation details.

Chapter 8 presents an evaluation of the case implementation, focusing on which SDA is most suitable for small business ecommerce SOA migration by incremental service integration. This evaluation is conducted based on the criteria found and described in chapter 5.

Chapter 9 presents a discussion of the case implementation results and evaluation findings.

Chapter 10 presents the conclusion of the report and its overall outcome.

Chapter 2

2 Background

This chapter briefly introduces the reader to the concepts, principles and technologies of service-orientation and service-oriented architecture. This is followed by a short description on SOA migration.

2.1 Service Orientation paradigm

The service orientation paradigm “anonyms with service oriented design²” is a way of design, focusing on designing software component logic, as self-contained, independent, loosely coupled services, that communicates in a standardized manner, through a well-defined interface. These services should follow the service orientation principles, also known as SOA principles, in order to achieve service oriented architecture [6].

2.1.1 SOA defined

Different definitions of SOA exists, some are more technically focused while others focus on the business perceptive.

This thesis will use a working definition adopted from a technical definition by Thomas Erl³: “SOA is a form of technology architecture that adheres to the principles of service orientation, when realized through the Web service technology platform; SOA establishes the potential to support and promote these principles throughout the business process and automation domains of an enterprise [7].”

2.1.2 SOA principles

Service-oriented principles are commonly known as SOA principles and serve

² Wiki: <http://en.wikipedia.org/wiki/Service-orientation>

³ Thomas Erl: SOA author and Series Editor of the "Prentice Hall Service-Oriented Computing Series ":
<http://www.thomaserl.com/>

guidelines for architects and developers in defining services in a service-oriented context.

Even though no official principles are defined for service-orientation, there exist a widespread accepted set of principles related to service orientation, considered core to the design of service regardless of underlying technologies, which have been found in the prior pre-thesis investigation [8]. These can be summed up to as follows:

- **Services are reusable** — Services are designed to support potential reuse.
- **Services share a formal contract** — For services to interact, they only need to share a formal contract that describes each service and defines the terms of information exchange.
- **Services are loosely coupled** — Services must be designed to interact without the need for, tight cross-service dependencies.
- **Services abstract underlying logic** — The only part of a service that is visible to the outside world is what is exposed via the service contract. Underlying logic, beyond what is expressed in the descriptions that comprise the contract, is invisible and irrelevant to service requestors.
- **Services are composable** — Services may compose other services. This allows logic to be represented at different levels of granularity and promotes reusability and the creation of abstraction layers.
- **Services are autonomous** — The logic governed by a service resides within an explicit boundary. The service has control within this boundary and is not dependent on other services for it to execute its governance.
- **Services are stateless** — Services should not be required to manage state information, as that can impede their ability to remain loosely coupled. Services should be designed to maximize statelessness even if that means deferring state management elsewhere.
- **Services are discoverable** — Services should allow their descriptions to be discovered and understood by humans and service requestors that may be able to make use of their logic.

2.1.3 SOA participants

The participants of a SOA are a service provider and a service requestor. The provider exposes the service by publishing its contract in a service registry where the service requestor can find/discover it, in order to determine how to bind and communicate with the service. Further communication is done directly between the requestor and provider.

This is illustrated on figure 1:

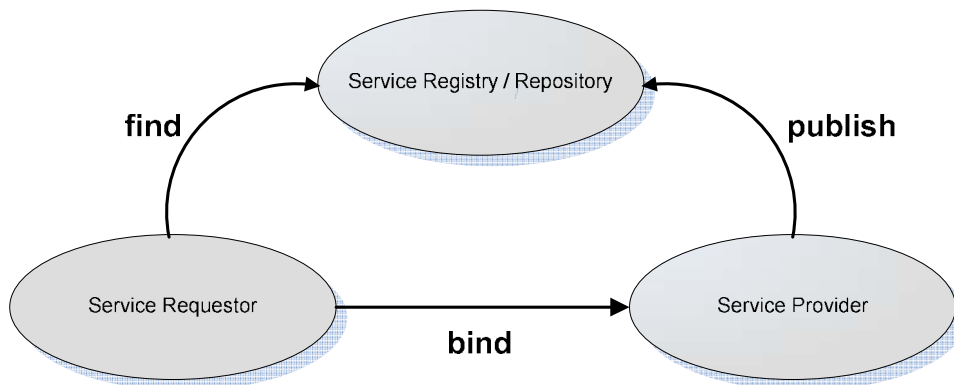


Figure 1: SOA participants

2.1.4 Web services as SOA enablers

Web services are software services that are self contained applications, which perform functions, for requests. The “Web service” model conforms at a minimum, to the Web service platform core specifications shown in figure 2, and is the primary method of implementing SOA [7].

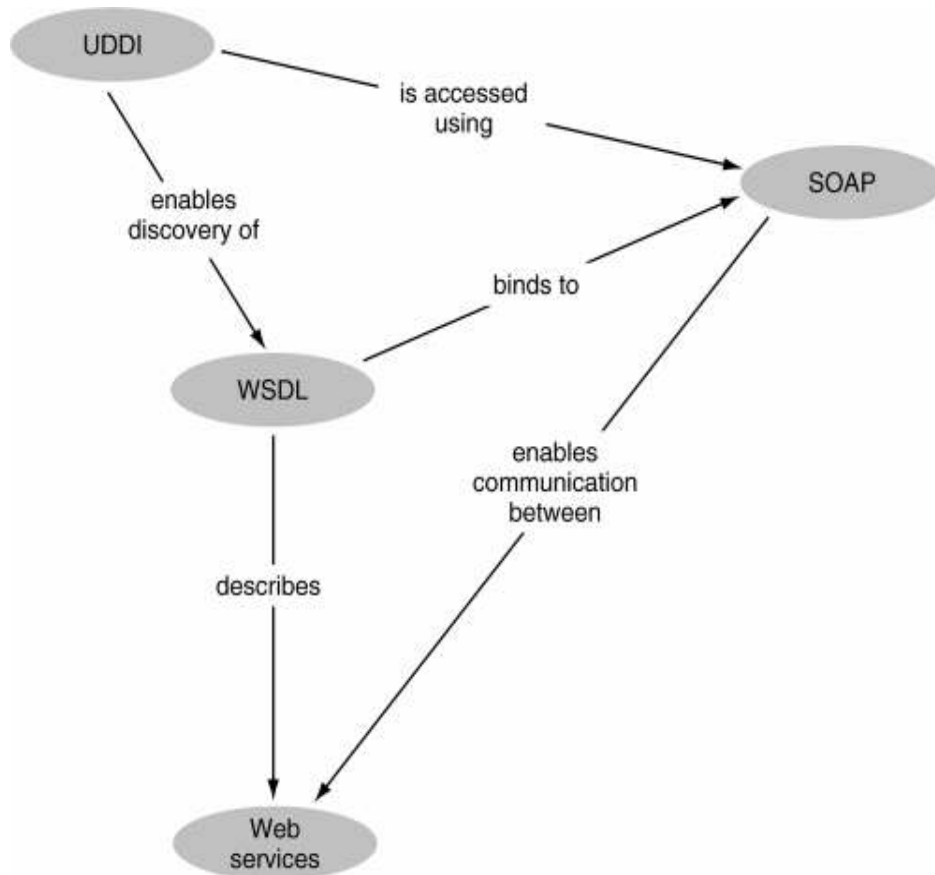


Figure 2: Web service model

WSDL is the standard format for describing a Web service. A WSDL definition describes how to access a Web service and what operations it will perform [9], and serves as a contract between the service provider and service consumer.

SOAP is a standard for exchanging XML-based messages over a computer network, normally using HTTP. SOAP forms the foundation layer of the Web services stack, providing a basic messaging framework, which more abstract layers can build on [10].

UDDI is a XML-based protocol that provides a distributed directory that enables businesses to list themselves on the Internet and discover other Web services [11].

2.1.5 Why SOA

The concept of SOA is a platform of self-contained, independent, loosely coupled services executing alone or composed with other services to fulfil business requirements, implementing business processes.

Further these services may be reused and reorganized to implement new business processes, as the markets and business needs may change.

It is stated that the reuse of services will lower the time and cost of new business process development, and that service interoperability and their loosely-coupled characteristic will increase flexibility and lighten the integration process with other businesses.

Carnegie Mellon[®] Software Engineering Institute (SEI) examined the quality attributes of SOA and states that the maturity of SOA have reached a point where the technology can help fulfil SOA promises of interoperability improving the possibilities of integration, extensibility and Modifiability, which businesses of all size may benefit from [12].

2.2 Migration

Information system migration moves the information system to a more flexible environment, while retaining the original system's data and functionality [13]. Architectural Migration focuses on migrating from one system architecture to another in this case a service oriented.

The need for migration comes from businesses who normally already employ some kind of information system, but wishes to gain benefits of a new environment and/or architecture, and is usually desired for systems that may pose difficulties in a number of area such as inflexibility, isolation, non-extensibility, lack of openness, etc. [14].

The essential idea is to allow businesses to preserve the current business logic, while at the same time transitioning into a new environment and/or architecture.

Some migrations are fully planned and the migration is executed in one big step. Other migrations are incremental and come from realizations of small project feedback, showing positive results.

In regards to the later, Edward Cobb, VP architecture and standards at BEA in regards to SOA implementation states: "It's all about starting small so you can learn how to monitor the benefits and see how your business is becoming more flexible and adaptable" [15].

This is backed up by Mark Prichard, product marketing director at BEA: "The benefits of SOA grow over time... through a series of small incremental wins" [15].

2.2.1 SOA migration

SOA migration is an architectural migration from any non-SOA system to a system that follows the service-oriented principles, in order to achieve service-oriented architecture.

In order for architectural migration to SOA, it is necessary to follow the SOA principles when identifying and implementing services.

Identification of functionality to be integrated into services is one of the essential steps when migrating to SOA, as the functionality must have a high level of importance in order to insure that the new service(s) will be used in many different project scenarios.

This is no simple task as many free open source ecommerce systems in the are written in scripting language such as PHP, PEARL, etc., with some of them being lesser concerned with code separation. Therefore a large amount of energy may be put on understanding the system at hand.

Chapter 3

3 Small business ecommerce

This chapter will discuss small businesses, ecommerce and central aspects for small businesses planning to migrate to SOA.

3.1 Small businesses:

Small businesses are generally businesses of a limited size and revenue. Nevertheless they have the benefits of dynamic decision making, as there do not exist, the same extent of hierarchy approval as large enterprises. In spite of this dynamicity being present, the decision making is in cases limited by strict budgets.

Small business generally don't have the luxury of large capital reinvestments in the companies IT or employing new information systems, and therefore place there focus on trying to utilize the present information systems to the fullest extent.

3.2 Adoption of ecommerce

Ecommerce has its history beginning in the late 1970's where it was mostly seen as a way of providing transaction electronically, through technologies as Electronic Data Interchange (EDI) and Electronic Fund Transfer, for used e.g. for purchase order or invoice exchange, between the companies and banks [16].

After the public introduction of the Internet in 1994 and further the development of HTTPS in 1998 the ecommerce term expanded to the selling, providing and buying goods and services over electronically most commonly over the World Wide Web (WWW).

Ecommerce were adopted by a large number of companies who saw the opportunity of conducting business anywhere, any time, reaching new markets. Furthermore the adoption made it possible for companies to project virtual storefronts to the world, exposing their goods to the world [17].

The introduction e-commerce has given small business an incentive to participate on the www, as the barrier to entry on market, is lowered significantly.

3.3 Importance of exposure

It is generally the case that exposure of the businesses products or services is of great importance.

Exposure makes customers aware of the business, its products or services, which can be translated to a probability of a buy and thereby an increase in revenue.

Different kinds of exposure are possible, such as mass-media, personal contact of possible prospects, etc... All of these exposure methods are expensive and not always cost possible, especially for small business.

One area that small business have found extremely effective and cheap to expose themselves is through the www, e.g. by homepages, and internet directories.

3.4 Current small ecommerce business:

Small ecommerce businesses utilize site catalogue, and search functionality in order to expose the companies' products or services.

The products or services are normally presented in a simple manner and ecommerce sites generally provide a search functionality. The standard ecommerce workflow from a customer's perspective is shown in figure 3, and gives a general idea of the ecommerce system functionality: [18]

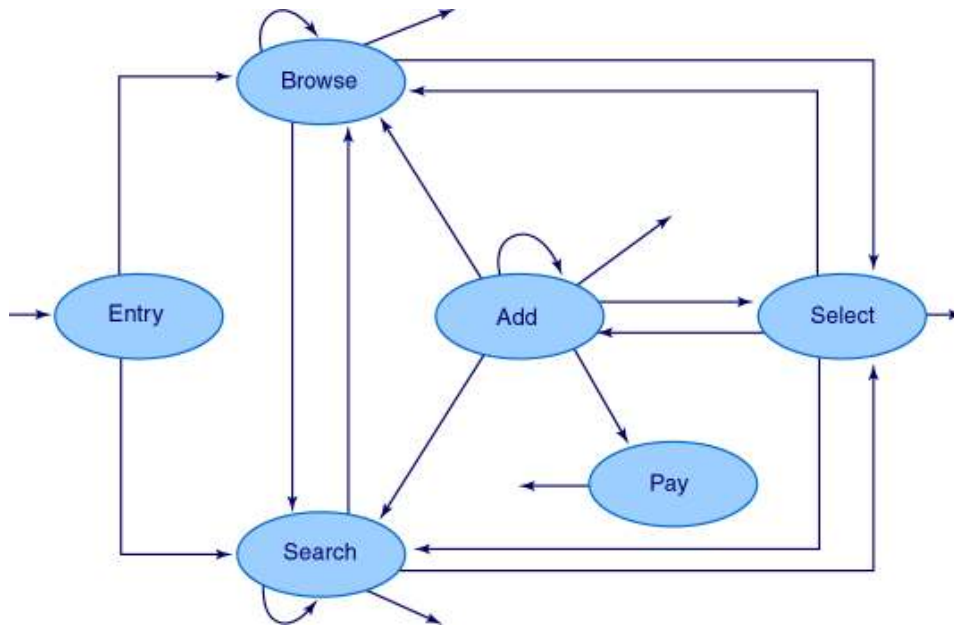


Figure 3: Online shopping flow

After entering a site, the next workflow states are to browse or search the business's catalog of products or services. A product-catalog is one of the core essential of any ecommerce businesses regardless of size. Searching the product catalog will therefore be the central case in the case implementation, described in chapters 6 and 7.

3.5 Important aspects in migrating to SOA

Small businesses have the important advantage of being able to change and take decisions faster. This in it self sets them closer to the goal of deciding to migrate to SOA. Nevertheless there are some aspects that need to be present in the businesses when migrating to SOA [19], [20], [21].

Capacity:

It is important to have the appropriate amount of employees with IT skills, and knowledge on development of SOA when wishing to start up a SOA migration project. This is a necessity, and any SDA which required a high amount of capacity will most likely be rejected before it is even started by small businesses.

Development Time:

It is important for business planning to commence a SOA migration project, to be able to allocate the necessary time when venturing into SO development process, as software development can be time consuming.

Cost:

Last but just as important is the capital to invest in such a project. The businesses must be willing to invest and be prepared that the cost can be higher than initially expected, if the any of the two above aspects are lacking.

Chapter 4

4 Literature review

The next section will present a literature review of different SDA for SOA migration.

4.1 Scope of the review

The literature review is presented to give insight in present research related to different legacy system to SOA (Service) migration approaches. This will set the stage for identifying the best suited SDA for non-SOA systems in small ecommerce businesses.

4.2 Keeping existing app. infrastructure

Existing applications hold a large amount of valuable functionality and business logic which must be considered when wanting to adopt SOA. Bisbal stated that legacy information systems are often the backbone of business and critical for business process execution and goes on to examine the impact of different modernizations techniques [14].

Quocirca stated that a complete replacement of existing infrastructure to adopt SOA is generally not viable and that the applications must participate in the new environment as they support the important business processes of the companies [22].

Quocirca further specified that for adopting SOA, existing applications must first be examined to determine which functionality is supporting these processes and which are replicates or could be trimmed away [22].

Zhan & Yang stated that legacy systems are valuable assets for the organizations, and that in a rapid changing business environment, Web services and SOA can help facilitate legacy system web enablement and transition into the service oriented environment [23].

4.3 Integrating legacy systems in services

For the approach of reengineering legacy system and assisting legacy code extraction for Web service construction in respect to, integrating legacy system in services for architectural migration to SOA Zhan & Yang [23] presented 3 approaches;

1. Legacy system to service integration via adapters by means of wrapping where only the legacy interface is analyzed.
2. Development of service(s) from scratch based on the business logic extracted from the legacy system through reverse engineering.
3. A hybrid which focuses on uncovering valuable business logic from the legacy code through reverse engineering of code and wrapping parts of the legacy system code to achieve valuable and agile service(s) for valuable business logic in a rapid and dependable way.

Zhan & Yang argued that the third approach can bridge the gap of effective migration of legacy system to service oriented architecture and implementation, and by allowing parts of the legacy system to become independent and distributable, at the same time keep the development costs down and utilizing the important business logic of the already familiar legacy system [23].

Zhan & Yang proposed an approach according with this, which uses a hierarchal clustering algorithm to identify possible service candidates [23].

Canfora, et al., on the other hand proposed a black box migration technique which does not require reverse engineering or code modification but deals with wrapping the original system through an exposure of a modernized interface in this case a Web service interface. The wrapper layer acted as an interpreter for an FSA machine that is used as a glue component in deciding how to transition into the next form in the legacy system or perform the next interaction action [24].

Lewis, et al., Further specifies that legacy component to service transformation by wrapping may be a straightforward discipline in specifying and setting Web service interfaces, to receive SOAP messages, parse their content and invoke legacy code based on it. Furthermore they stated that this approach is generally attractive for businesses venturing into the SOA migration space. This method is

further eased by current development tools, which help in the creation process [25].

Even though this might be the case Lewis, et al., stated that different characteristics of a legacy system, can complicate this process, and that analysis of feasibility and effort involved may need to be addressed first [25].

4.4 Summary

Existing application are seen as valuable assets, and should not just be discarded. They hold important functionality that can be incorporated into services when migrating to a SOA.

3 approaches of system integration in services were identified:

Wrapping is seen as the fastest easiest way to transform existing applications to services, but also as a short term solution that can complicate future maintenance. Redevelopment is seen as the most revolutionary approach but also the most intrusive. A method in between of partial component code reuse in services is seen as a plausible way.

The literature is good when it comes to explaining the different steps that must be followed to achieve migration from legacy code to services and SOA, but comes short on the importance of, giving argument based explanations, why one method has been chosen over the other.

Which approach of service development will be best suited in respect to small businesses ecommerce systems when migrating to SOA by incremental service integration. This is further examined in the next chapters.

Chapter 5

5 SDAs for SOA migration

The literature review, uncovered three main SDA for migration to be used when migrating to SOA, which will be referred to as the Wrapping, Redevelopment, and Hybrid approach.

Furthermore even though the approaches focuses on legacy system migration to services as their primary source of application, the approaches are seen feasible in the case of non-SOA systems, which do not follow the definition of legacy system.

5.1 Wrapping approach

The approach of wrapping, just at it name implies focuses on wrapping the system code preferably in component form with a service interface, enabling it to act as service. The wrapping approach focuses its attention on studying the interface of the code or component in order to replace this by a service interface, usually a Web service interface described by a WSDL service contract. This is seen as the quickest approach an in the literature reviewed as the preferred approach. Though this is the case, different statements state that this approach may complicate future maintenance and development, because it does not focus on deep knowledge original system in details before implementing the service/s. This is seen as a short time solution by some.

The downside to this approach is that it does not focus on the identifying system logic by reengineering and requires the system as a whole to be wrapped. This is not suited for incremental migration. Therefore this approach will not be examined in the case implementation.

5.2 Redevelopment approach

This approach focuses on reengineering the system, or parts of it, at some given level, in order to identify important business logic in the system code. This code must then be studied and new services are may be implemented based on the logic extracted. Here the actual code of the system is used as guidance and, new

code can be implemented as the developer sees best fit to fulfil the given requirements. This method requires good insight in the system to be migrated and study of the structure in order to identify and separate out a given section of logic. Zhang & Yang stated that this method “enables the developer to implement the service using new Web service techniques and languages, saves maintenance and improves efficiency for transaction, but it is highly invasive and that it is difficult to recover business logic perfectly” [23].

5.3 Hybrid approach

This approach is a hybrid between the two approaches described above. The Hybrid approach incorporates the notion of reengineering parts of the system in order to extract import code before wrapping it. Zhang & Yang states, that this approach is practical [23], because the code is already present, and good overview of the system for future development and maintenance is obtained.

5.4 Migration approach evaluation criteria

This thesis, evaluates 2 of the 3 identified SDAs described above, feasible for incremental architectural SOA migration. Evaluation will be conducted based on the criteria in the schema below.

The following criteria for evaluation of the different approaches have been chosen, based on their importance to software development, in environments where capacity, cost and time have high impact on the outcome of the implementation, in this case small ecommerce businesses.

Chapter 5 SDAs for SOA migration

	Description	Important Criteria for evaluating approaches for SMB EC	Related to aspect
Overall development complexity.	The level of complexity that the given approach will inflict on the development process. E.g. Redevelopment could require integration of outside components and complex setup steps in order to implement the service.	Complexity normally increases development time and requires capacity to be present.	Capacity, Time, Cost.
System source code understanding.	How much knowledge do the developers need to have or obtain about the system source code, in order to implement the service using the given development approach.	Company systems, are usually not implemented by them self. In the case of small ecommerce business it is commonly an out of the box solution, usually open source and free. Therefore, there do not always exist, a deep knowledge of the system source code (implementation). To obtain this requires time, capacity, cost.	Capacity, Time, Cost.
Ease of Extensibility	How easy is it to incorporating additional capabilities into existing services or add new ones to the system.	Ease of extensibility is a necessity for small business, as capacity and time is sparse. Small business like flexible solutions that can easy be extended.	Capacity, Time, Cost.
Dependencies	Amount of dependency on software components which are not part of the specific service.	Low dependency on the existing system means greater flexibility in new development. It is necessary to minimize the service dependencies on the source system. This will reduce future development time.	Capacity, Time, Cost.
Effort to achieve service principles	How Much effort is needed to achieve a service that follows the SOA principles.	In order to migrate to SOA it is important to have the services follow the SOA principles in order to achieve a SOA system. This is especially important for the businesses that do not	Capacity, Time, Cost.

		have the capacity, time or cost to remake services.	
--	--	---	--

Table 1: Evaluation criteria

Chapter 6

6 Application of approaches

This chapter deals with the application of the identified approaches in chapter 5 to a case, on an open source ecommerce system (OSCommerce) in order to evaluate and compare the different approaches in respect to criteria found in previous chapter.

6.1 Purpose

The purpose of this case implementation is to acquire results regarding which approach is best fitted for an ECommerce system used by small business in regards to incremental SOA migration. For this a free open source system OSCommerce is used.

The basic idea of the case is to implement two separate and independent services by means of Web service technology. Hereafter an evaluation of the development processes will be carried out, evaluating and comparing the different approaches in respects to the criteria found in chapter 5.

6.2 Limits

The Wrapper approach described in chapter 5.1 will not be implemented in the case. This is due to the fact that it promotes wrapping of the entire system, where this thesis is interested in, results based of system integration in service, when *incrementally* migrating to SOA.

Therefore the Hybrid approach and Redevelopment approach described in section 5.2 and 5.3 respectively will be the centre of this case implementation.

Furthermore as the architecture of the rest of the system is not SOA, and the focus is not on the implementation of SOA as much as, it is on SDAs for incremental SOA migration, in this thesis, the focus will be on implementing an independent service.

6.3 Requirements analysis

This Section deals with the requirements analysis. It will present business goals, system of interest and use case description.

6.3.1 Business goal & stakeholders

This thesis examined in chapter 3 the importance of exposure for small businesses. This has been decided as the main goal of the implementation. The stakeholder is the owner of the ecommerce system who wishes to expose his products, for future use in e-marketplaces, through services. This may open up for new sales channels and collaboration opportunities.

6.3.2 Open source ecommerce

The system to be utilized for service integration is an Open Source Ecommerce system, OSCommerce⁴ that claims they have over 12.000 registered shop users and a community of over 150.000.

The reasons why open source system has been chosen is because it is free under the GNU, and is therefore presumably be utilized by small ecommerce businesses, focusing on cost as on of their primary criteria.

Furthermore it would be difficult to obtain the source code for a major commercial ecommerce system, and therefore this is not interesting for the thesis which addresses how small business could best migrate to SOA, by own initiative.

6.3.3 Use case

A use case is initiated by a user with a particular goal in mind, and completes successfully when that goal is satisfied. It describes the sequence of interactions between actors and the system necessary to deliver the service that satisfies the goal. It also includes possible variants of this sequence, e.g., extensions (alternative flows) sequences that may also satisfy the goal, as well as sequences that may lead to failure to complete the service because of exceptional behaviour, error handling, etc. The system is treated as a "black box", and the interactions

⁴ <http://www.oscommerce.com/>

with the system, including system responses, are as perceived from outside the system.

Use Case 1:

Primary Actor: Service Consumer (Human).

Stakeholders and interests:

- Consumer: Wants to view Search products of the given ecommerce site.

Preconditions: Consumer has developed client that can communicate with Web service and display product information.

Success Guarantee (Post): Product information and price is displayed.

Main Success Scenario:

1. Service Consumer enters product to search for.
2. Service searches system for given product.
3. Service sends information on product back to Service Consumer.
4. Service Consumer gets product information and price displayed.

Extensions (or Alternative Flows):

- 2a. No product found.
1. Service sends empty value back.
 2. Service Consumer can search again.

Related information:

Priority: High

Frequency: Very often, many times

6.4 Requirements specification

6.4.1 System

- The Open Source Ecommerce system, OSCommerce have been decided upon for the actual implementation.

6.4.2 Approach

- The service must be implemented in using both the Hybrid approach and the Redevelopment approach, described in section 5.2 and 5.3.

6.4.3 Service

- The Service consumer must be able to search for shop products through the service and get results back.
- Input must be a single search parameter.
- The result sent back should contain product name, and price
 - Name of product
 - Price with tax
- The service must offer a shared formal contract, described by a WSDL document and communicate using SOAP.
- The service must follow the SOA principles to the extent this is possible.

6.4.4 Software

- The hybrid service must be implemented in PHP, as the rest of OSCommerce is.

- NuSOAP PHP library for Web service development must be used for the implementation of the Hybrid service.
- Microsoft Windows must be installed, along with OSCommerce, PHP and APACHE Web Server.
- The Redevelopment service must be implemented in Java.
- Apache Axis should be used to facilitate Java Web service functionality.

6.5 Design

6.5.1 Overall design

The Overall design of the implementation of the 2 approaches is shown in figure 4

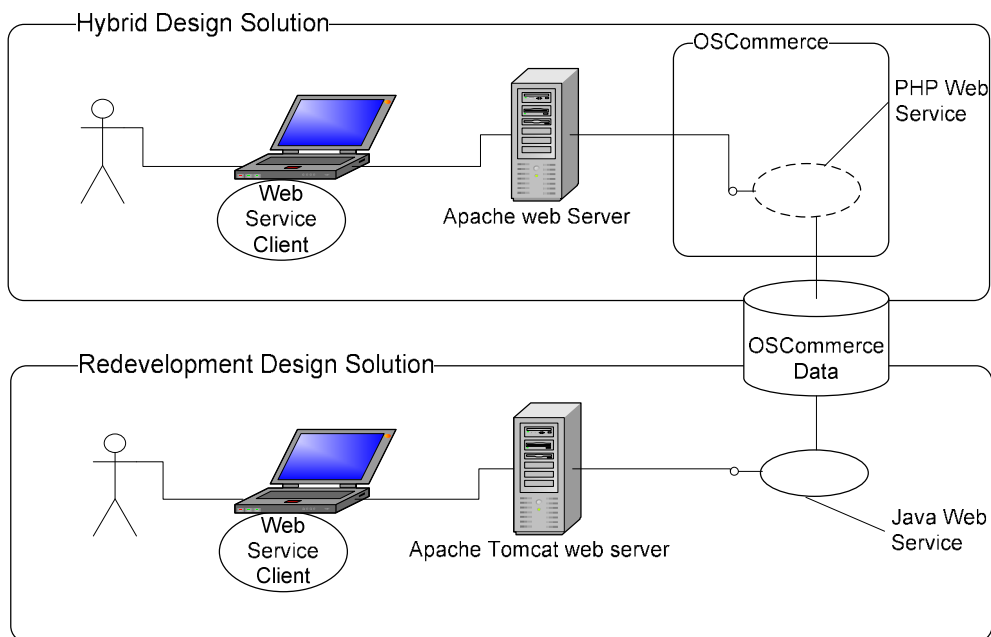


Figure 4: Solution Designs

The top of figure 4 shows the Hybrid approach design. A Web service will be developed based on identifying the code already present in OSCommerce for searching the product catalogue. This code will be wrapped with a Web service

layer by implementing it as a function and describing how to communicate with the function using WSDL, and registering it as a service, using appropriate PHP Web service library. This allows users outside the system to access the product catalogue by utilizing the service.

The bottom section of figure 4 shows the Redevelopment approach design. Here a Java Web service and its logic will be implemented from scratch based on the logic for searching the product catalogue. This allows users outside the system to access the product catalogue by utilizing the service.

Both design solutions use the OSCommerce tables in the database, but where as the first solution makes use of OSCommerce specific database calls as it wraps the functionality already present, the latter described approach uses a J/Connector and standard MYSQL calls to contact the database.

6.5.2 OSCommerce product search functionality

The OSCommerce product search functionality builds a SQL statement based on the input parameters received from the browser and uses the statement to query the OSCommerce tables for information for specific product or products in a product category. The code for the product search functionality code can be inspected in appendix B.

6.5.3 Service design

The design of the services developed by the 2 approaches will be described below. Both services are based on or inspired by the OSCommerce product search functionality described in 6.5.2.

Interface:

The service takes the name of a specific product or product category to search for in the product catalogue as input parameter, and returns the name and price of the specific product or a list of names and prices for a given product category.

In the first case where a specific product is search for e.g. a movie called "A Bug's Life" the following will be returned:

A Bug's Life	35.99
--------------	-------

Figure 5: Single result

An e.g. of a search by a product category, would return the result seen below from an input of the product category "DVD":

A Bug's Life	35.99
Blade Runner - Director's Cut	30.00
Courage Under Fire	29.99
Die Hard With A Vengeance	39.99

Figure 6: Multiple results

Format:

The format chosen in is a very simple one. A string is given as input and a comma separated string as output which can be processed by clients or other services as needed.

Input (String: search): "DVD"

Output (String: return): "A Bug's Life 35.99,Beloved 54.99,Blade Runner - Director's Cut 30.00"

Hybrid Service:

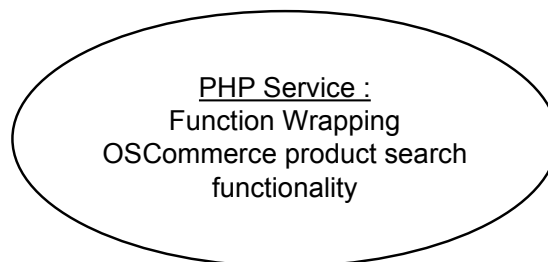


Figure 7: Hybrid service

In the Hybrid service solution a function will act as a wrapping layer encapsulating the logic required for searching the shops product catalogue. The function is registered as service and its interface is described by a wsdl document.

Redevelopment Service:

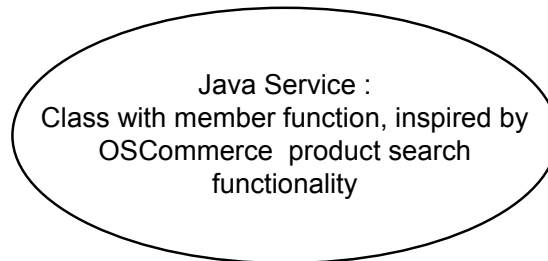


Figure 8: Redevelopment service

The Redevelopment service is a class, which is registered on a application server. It is based on Java and Apache Axis to implement the Web service. The member function which fulfils the requirement of delivering product information based on search criteria is here independent of the OSCommerce system, with logic implemented from scratch, though it takes inspiration from the product search functionality, by utilizes the same SQL statement generated in the hybrid service.

It takes the same argument as the Hybrid service described above and returns the same.

The actual code for each of the two services can be found in appendix C.

6.5.4 Contract

Both services will be described by a WSDL document to enable consumers of the service to communicate with the respectable service through the service interface. The WSDL document for each service will be auto generated using Web Service tools for PHP and Java.

The structure of the WSDL document follows the WSDL 1.1 specification [26] and can be seen here below in listing 1. It contains the service description elements and defines a reference to the Web service description:

```
<definitions>
  <types>
    <!-- which provides data type definitions used to describe the
         messages exchanged. -->
  </types>
  <message>
    <!-- represents an abstract definition of the data being
         transmitted. A message consists of logical parts, each of which
         is associated with a definition within some type system. -->
  </message>

  <portType>
    <!-- a set of abstract operations. Each operation refers to an input
         message and output messages. -->
  </portType>
  <binding>
    <!-- specifies concrete protocol and data format specifications
         for the operations and messages defined by a particular
         portType. -->
  </binding>
  <service>
    <!-- used to aggregate a set of related ports. -->
    <port>
      <!-- specifies an address for a binding, thus defining a single
           communication endpoint. -->
    </port>
  </service>
</definitions>
```

Listing 1: WSDL format

The Actual WSDL documents for each of the services can be shown in the implementation chapter 7.

6.5.5 Communication and protocol

Communication with the Web services described above will be accomplished by using SOAP as seen on figure 9.

The services use SOAP RPC-style messaging to communicate, by using remote procedure calls that are encoded in xml and wrapped in a SOAP envelope, by

request-response interaction, where the client sends a SOAP message containing argument values to a method.

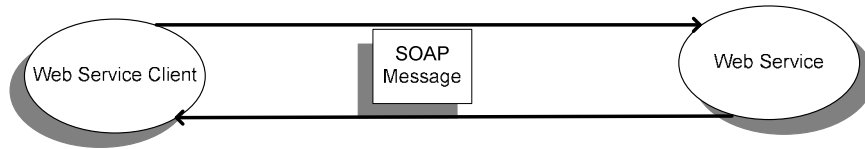


Figure 9: SOAP communication

The structure of the SOAP message follows SOAP 1.1 and can be seen in listing 2.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
  <SOAP-ENV:Envelope SOAP-ENV:encodingStyle=
    "http://schemas.xmlsoap.org/soap/encoding/">
    <SOAP-ENV:Header>
      <!-- Can Contain meta data about how the message should be
        processed by SOAP nodes -->
    <SOAP-ENV:Header>
    <SOAP-ENV:Body>
      <!-- Comprises WDSL message elements and name of operation to be
        executed -->
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
  
```

Listing 2: SOAP message structure

E.g. for the Hybrid service the request-response SOAP envelope will look like the following listings 3 and 4, with `prodSearch()` being the method to execute, with the input string "A Bug's Life" and the output returned being "A Bug's Life 35.99,".

Request

```

<?xml version="1.0" encoding="ISO-8859-1"?>
  <SOAP-ENV:Envelope SOAP- ENV:encodingStyle=
    "http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  >
  
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="urn:productSearch">
  <SOAP-ENV:Body>
    <tns:prodSearch xmlns:tns="urn:productSearch">
      <search xsi:type="xsd:string">A Bug's Life</search>
    </tns:prodSearch>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope
>
```

Listing 3: SOAP request example

Response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle=
"http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <ns1:prodSearchResponse xmlns:ns1="urn:productSearch">
      <return xsi:type="xsd:string">A Bug's Life 35.99</return>
    </ns1:prodSearchResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Listing 4: SOAP response example

Chapter 7

7 Implementation

This Chapter deals with the implementation of the 2 services described in the previous design section.

7.1 OSCommerce implementation

In order to implement the two Web services, OSCommerce had to be implemented first. PHP, Apache Web Server and MYSQL have to be installed. For this JSAS was employed.

7.1.1 OSCommerce structure

OSCommerce is an Open Source Ecommerce system, for use under the GNU License agreement. OSCommerce is implemented in PHP, using a MySQL database Server to implement its data layer. OSCommerce Version 2.2rc1 is used.

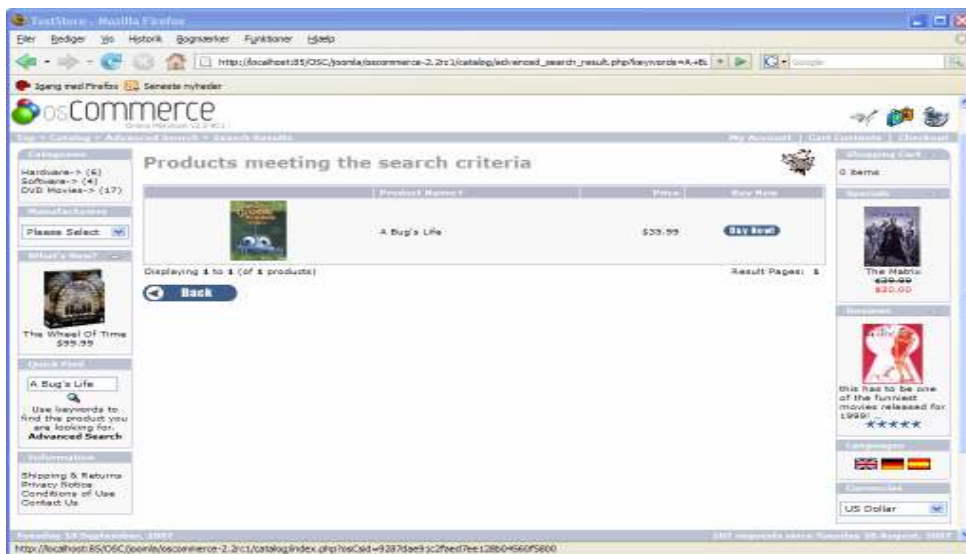


Figure 10: OSCommerce shop

To use OSCommerce, the Joomla zip file were replaced the OSCommerce zip file before installation.

The JSAS version used in this project includes the following:

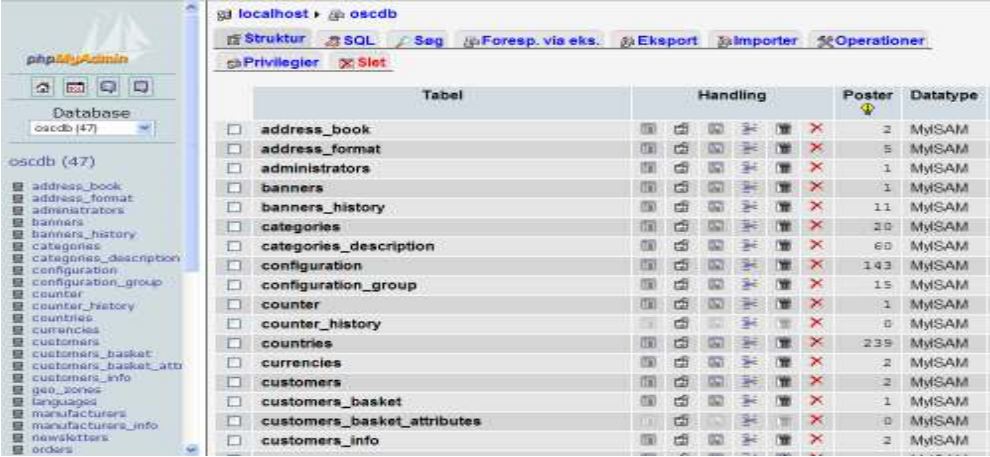
- JSAS 1.10
 - Apache HTTP Server 2.0.54
 - MySQL Server – 4.1.14-nt
 - PHP Version 4.4.4
 - phpMyAdmin – 2.8.2.4

Upon installation of OSCommerce, a database is created on the MySQL database server with necessary OSCommerce tables.

The OSCommerce database has been given the name of **oscdb**. Any OSCommerce call to the database will use the oscdb as the default database if nothing else is specified.

phpMyAdmin which also included in JSAS is a administration tool written in PHP for handling MySQL over the Web⁶.

Figure 12 shows an excerpt of the OSCommerce database and tables seen through the phpMyAdmin interface:



Tabel	Handling	Poster	Datatype
<input type="checkbox"/> address_book		2	MyISAM
<input type="checkbox"/> address_format		5	MyISAM
<input type="checkbox"/> administrators		1	MyISAM
<input type="checkbox"/> banners		1	MyISAM
<input type="checkbox"/> banners_history		11	MyISAM
<input type="checkbox"/> categories		20	MyISAM
<input type="checkbox"/> categories_description		60	MyISAM
<input type="checkbox"/> configuration		143	MyISAM
<input type="checkbox"/> configuration_group		15	MyISAM
<input type="checkbox"/> counter		1	MyISAM
<input type="checkbox"/> counter_history		0	MyISAM
<input type="checkbox"/> countries		239	MyISAM
<input type="checkbox"/> currencies		2	MyISAM
<input type="checkbox"/> customers		2	MyISAM
<input type="checkbox"/> customers_basket		1	MyISAM
<input type="checkbox"/> customers_basket_attr		0	MyISAM
<input type="checkbox"/> customers_info		2	MyISAM
<input type="checkbox"/> languages		1	MyISAM
<input type="checkbox"/> manufacturers		0	MyISAM
<input type="checkbox"/> manufacturers_info		0	MyISAM
<input type="checkbox"/> newsletters		0	MyISAM
<input type="checkbox"/> orders		2	MyISAM

Figure 12: phpMyAdmin showing oscdb

⁶ http://www.phpmyadmin.net/home_page/index.php

The OSCommerce system has been installed to the following Virtual folder: W:\ under www\OSC\joomla\oscommerce-2.2rc1\ with the catalog folder as a subfolder at W:\ under www\OSC\joomla\oscommerce-2.2rc1\catalog.

7.2 Hybrid service implementation

The Hybrid service has been implemented in PHP to accommodate the PHP logic already part of OSCommerce search functionality.

7.2.1 NuSOAP library

NuSOAP is a group of PHP classes that allow developers to create and consume SOAP Web services. It does not require any special PHP extensions, and supports much of the SOAP 1.1 [27] specification. It can generate WSDL 1.1 and also consume it for use in serialization [28]. NuSOAP is used in the implementation of the hybrid service.

This is done by placing the NuSOAP library in the catalog folder and referencing in the productSearch.php file implementing the Web Service, located in the catalog directory. The NuSOAP was used for automatically generating the WSDL file for the given Web service.

7.2.2 Code excerpt

The following code excerpts will go through important parts of the service implementation:

```
<?php
// Pull in the NuSOAP code
require_once('lib/nusoap.php');
// Create the server instance
$server = new soap_server();
// Initialize WSDL support
$server->configureWSDL('productSearch', 'urn:productSearch');
// Register the method to expose
$server->register('prodSearch', //method name
                array('search' => 'xsd:string'), //input
                array('return' => 'xsd:string'), //output parameters
                'urn:productSearch', // namespace
                'urn:productSearch#prodSearch', //soapaction
                'rpc', // style
```

```
'encoded', // use
'Searches OSCommerce product catalog and
returns result' // documentation
);
```

Listing 5: Registering service with NuSOAP

First the NuSOAP Library is referenced. `configureWSDL()` is used to support the generation of the WSDL document for the service `productSearch`. NuSOAP `register()` is used to register the function `prodSearch()` and parameters as part of the `productSearch` service.

```
//..
//..
/HMU ADDED WRAPPER START

function prodSearch($search) {
    //...
    //...
    $keywords = $search;
    //..
    //..
```

Listing 6: prodSearch defined

The `prodSearch()` function acts as the Web service wrapper. It sets the variable `$keywords` which is used by the OSCommerce search functionality to limit the search to given keyword/s

```
//..
//..
$select_str = "select distinct " . $select_column_list . "
m.manufacturers_id, p.products_id, pd.products_name,
p.products_price, p.products_tax_class_id, IF(s.status,
s.specials_new_products_price, NULL) as
specials_new_products_price, IF(s.status,
s.specials_new_products_price, p.products_price) as final_price ";
//..
//..
```

Listing 7: Select part of sql statemet

The select part of query statement is built, the same goes for the, from part, where part, and order part of the query statement as seen below.

```
//..
//..
```

```
$listing_sql = $select_str . $from_str . $where_str . $order_str;
```

Listing 8: Complete sql statement

The final sql statement is built.

```
//..  
//..  
/HMU ADDED WRAPPER ENDPOINT  
  
$listing_query = tep_db_query($listing_sql);  
$row = array();  
$rows = array();  
$i = 0;  
while($row = mysql_fetch_array($listing_query)){  
    // Make comma separated format  
    $rows[$i] = $row[3] . ' ' . trim(trim($row[7], '0'), '.');  
    $i++;  
}  
mysql_free_result($listing_query);  
  
$commaSeperatedResult = implode(",",$rows);  
return ($commaSeperatedResult);
```

Listing 9: fetching rows from appropriate oscdb tables

The while loop uses `mysql_fetch_array($listing_query)` to fill `$rows` name and price returned from OSCommerce tables. Finally the name/s and price/s are returned in a comma separated format, "Beloved 39.50, A Bug's Life 35.99"

From the excerpts it can be seen that the service `productSearch` implements the function `prodSearch()`. The implementation is located in the file `productSearch.php` in the `catalog` folder having the path: `W:\www\OSC\joomla\oscommerce-2.2rc1\catalog\productSearch.php` with the NuSOAP library located at path: `W:\www\OSC\joomla\oscommerce-2.2rc1\catalog\lib\`

7.2.3 WSDL

The WSDL document for the Hybrid service is generated by the going to the following path:

W:\www\OSC\joomla\oscommerce-2.2rc1\catalog\productSearch.php?wsdl. The NuSOAP, auto generates the WSDL document seen below in listing 10, based on the input parameters in to the `configureWSDL()` and `register()` functions.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<definitions xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="urn:productSearch" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="urn:productSearch">
<types>
<xsd:schema targetNamespace="urn:productSearch">
<xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
<xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
</xsd:schema>
</types>
<message name="prodSearchRequest">
<part name="search" type="xsd:string" />
</message>
<message name="prodSearchResponse">
<part name="return" type="xsd:string" />
</message>
<portType name="productSearchPortType">
<operation name="prodSearch">
<documentation>Searches OSCOMmerce product catalog and returns result</documentation>
<input message="tns:prodSearchRequest" />
<output message="tns:prodSearchResponse" />
</operation>
</portType>
<binding name="productSearchBinding" type="tns:productSearchPortType">
<soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
<operation name="prodSearch">
<soap:operation soapAction="urn:productSearch#prodSearch" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:productSearch"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:productSearch"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
</binding>
<service name="productSearch">
<port name="productSearchPort" binding="tns:productSearchBinding">
<soap:address location="http://localhost:85/OSC/joomla/oscommerce-
2.2rc1/catalog/productSearch.php" />
</port>
</service>
</definitions>
```

Listing 10: WSDL document for Hybrid service

As can be seen, the service name is productSearch, with a method called prodsearch, input being the string search, and output the string result. The service communicates using SOAP and the messaging style is rpc.

7.3 Redevelopment service implementation

The Redevelopment service has been implemented in Java and is inspired by the OSCommerce search functionality.

7.3.1 Tomcat Web server

The Tomcat server is a Java based Web application container and Web server that was created to run Servlets and JavaServer Pages (JSP) in Web applications [29]. Version 4.0.1 is installed in the path: C:\jakarta-tomcat-4.0.1\jakarta-tomcat-4.0.1

7.3.2 Apache Axis

Apache Axis is an open source, XML based Web service framework. It consists of a Java and a C++ implementation of the SOAP server, and various utilities and APIs for generating and deploying Web service applications [30].

Apache Axis 1.0 is used Together with Tomcat to develop and run the Redevelopment Web service.

Axis is downloaded and extracted in to the apache tomcat directory path: C:\jakarta-tomcat-4.0.1\jakarta-tomcat-4.0.1\webapps\axis

For complete setup steps please refer to appendix A.

7.3.3 Code excerpt

The following code excerpts will go through important parts of the service implementation:


```
import java.sql.*;
public class productSearch
{
//..
//..
```

Listing 11: Class productSearch

Implements a class called `productSearch`, which will acts as the Web service.

```
public String prodSearch(String search) {
    // Instantiate variables
    Connection con = null;
    Statement st = null;
    ResultSet rs = null;
    String result = null
;
}
```

Listing 12: Definition of variables

The Web Service method `prodSearch()` is defined.

```
try {
    // Get connection to oscdb using jdbc connector
    Class.forName("com.mysql.jdbc.Driver").newInstance();
    con = DriverManager.getConnection("jdbc:
        mysql:///oscdb","root", "");
    // Create and execute statement
    st = con.createStatement();
    rs = st.executeQuery("select distinct p.products_image,
        m.manufacturers_id, p.products_id,
        pd.products_name, p.products_price,
        p.products_tax_class_id, IF(s.status,
        s.specials_new_products_price, NULL) as
        specials_new_products_price, IF(s.status,
        s.specials_new_products_price, p.products_price) as
```

```
        final_price from products p left join manufacturers
        m using(manufacturers_id) left join specials s on
        p.products_id = s.products_id, products_description
        pd, categories c, products_to_categories p2c where
        p.products_status = '1' and p.products_id =
        pd.products_id and pd.language_id = '1' and
        p.products_id = p2c.products_id and
        p2c.categories_id = c.categories_id and
        ((pd.products_name like '%' + search + '%' or
        p.products_model like '%' + search + '%' or
        m.manufacturers_name like '%' + search + '%') )
        order by pd.products_name");

// Get rows
while(rs.next()) {
    result += rs.getString(4) + " " +
    rs.getString(8).substring(
    0,rs.getString(8).indexOf(".") + 3) + ",";
}
```

Listing 13: Fetch data from oscdb

Call to `oscdb` based on logic from OSCommerce search functionality is executed, rows retrieved and result strings formatted.

```
// Return result
return result
;
```

Listing 14: Return result

The product information is returned in the right format.

From the excerpts it can be seen that the service `productSearch` implements the function `prodSearch()`. The implementation is located at the path:

C:\jakarta-tomcat-4.0.1\jakarta-tomcat-4.0.1\webapps\axis\localhost\ProductSearch.class

7.3.4 WSDL

The WSDL document for the Redevelopment service is generated by going to the following path <http://localhost:8080/axis/productSearch.jws?wsdl>. Axis auto generates the WSDL document seen below in listing 15. The WSDL is very similar to the WSDL of the Hybrid service.

```
<?xml version="1.0" encoding="UTF-8" ?>
<wsdl:definitions targetNamespace="http://localhost:8080/axis/productSearch.jws"
  xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:apachesoap="http://xml.apache.org/xml-soap"
  xmlns:impl="http://localhost:8080/axis/productSearch.jws"
  xmlns:intf="http://localhost:8080/axis/productSearch.jws"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wSDLsoap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:types />
  <wsdl:message name="prodSearchResponse">
    <wsdl:part name="prodSearchReturn" type="xsd:string" />
  </wsdl:message>
  <wsdl:message name="prodSearchRequest">
    <wsdl:part name="search" type="xsd:string" />
  </wsdl:message>
  <wsdl:portType name="productSearch">
    <wsdl:operation name="prodSearch" parameterOrder="search">
      <wsdl:input message="impl:prodSearchRequest" name="prodSearchRequest" />
      <wsdl:output message="impl:prodSearchResponse" name="prodSearchResponse" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="productSearchSoapBinding" type="impl:productSearch">
    <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="prodSearch">
      <wsdlsoap:operation soapAction="" />
      <wsdl:input name="prodSearchRequest">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="http://localhost:8080/axis/productSearch.jws" use="encoded" />
      </wsdl:input>
      <wsdl:output name="prodSearchResponse">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="http://localhost:8080/axis/productSearch.jws" use="encoded" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="productSearchService">
    <wsdl:port binding="impl:productSearchSoapBinding" name="productSearch">
      <wsdlsoap:address location="http://localhost:8080/axis/productSearch.jws" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

Listing 15: WSDL for Redevelopment service

Chapter 8

8 Evaluation of case implementation

The two SDAs in the case implementation, is evaluated in regards to the criteria described in chapter 5.

8.1 Overall development complexity

Hybrid approach solution:

The Hybrid approach solution required the identification of the product search functionality and the Web service library NuSOAP, in order to implement the service. The product search functionality, already part of OSCommerce have been developed and tested before the release of OSCommerce and therefore presumed to work correctly. The attention was therefore directed on establish the boundaries of the service, by identifying the appropriate start and endpoint for the wrapper. This approach therefore was relatively simple to implement, with a low level of complexity.

Redevelopment solution

The Redevelopment approach solution required the identification of the product search functionality, a separate Web server (Tomcat), Axis Web service library and a database driver, in order to implement the service. The product search functionality had to be extracted, in order to develop the new service based on the old system logic. This required some code understanding. Further a database driver connector had to be set up and referenced by the new service in order to communicate with the OSCommerce product tables. This approach has a higher complexity level than the above.

Comparison

The Hybrid approach solution required fewer setup steps in order to begin implementing the service. It also required fewer components to be integrated, as these were already incorporated by the OSCommerce system. These factors

eased the development complexity in comparison to the Redevelopment approach solution where more steps and technologies had to be executed and integrated respectively, as well as the fact that code had to be written, after extracting the functionality, making the development a more complex effort.

8.2 System source code understanding

Hybrid solution:

While implementing the Hybrid approach solution service, it became clear that understanding of system source code was necessary. This related first to identifying the product search functionality and next to understanding the code just enough to determine where to place the service start and endpoint. Still in dept details of the actual search functionality was not necessary, but more the input and output parts of the code sections, which would be used as the wrapper start and endpoints.

Redevelopment solution

The Redevelopment approach solution service required knowledge of the system source code. First the product search functionality had to be identified. Next an in dept understanding of the search functionality was necessary in order to understand and extract the functionality. This was then used as the basis for developing the new service from scratch. Even tough a static solution for retrieving the product data for OSCommerce products was employed to simplify the service; the decision was based on a clear understanding of the functionality of the code.

Comparison

Both solutions required some system code analysis in order to find the search functionality. Nevertheless the Redevelopment approach solution goes into more details in code analysis, in order to determine how the development of the new service can best be designed and implemented. Therefore a higher level of understanding of system source code is necessary in the Redevelopment approach solution.

8.3 Ease of extendibility

Hybrid solution:

The Hybrid approach solution is developed to be independent of any new service implementation, by communicating through operations specified by a WSDL document. This allows for ease of further system extendibility, where new services can be integrated without taking the Hybrid solution service into concern.

Nevertheless when it comes to adding additional capabilities to the existing service it will require in dept knowledge of the functionality of the service. This is not the focus when wrapping the partial system code and therefore will make the extension more difficult. Further difficulties can occur as the service is based on already developed code, which is closely related to the specific functionality, and therefore increase the possibility of the new functionality clashing with the old.

Redevelopment solution

The Redevelopment approach solution also allows for ease of system extendibility. It allows for new services to be developed and communicate with the service as it also presents an interface for communication through WSDL. When it comes to service capability extension there is a higher possibility of the developers already having knowledge of the service functionality as it is developed from scratch. Further the low dependency on the system source code also eases the future extension of the service.

Comparison

Both solutions though using different approaches makes it easy to extend the entire system with more services. This is because both approaches focus on building services where logic is encapsulated and contained within the service interface. When it comes to extending the services the Redevelopment solution approach is preferred.

8.4 Dependencies

Hybrid solution:

If the system is to be updated, there is a chance that the functionality will also change, especially in an open source systems; this means that code in the service must be re-factored. This is a dependency on the source system. Furthermore if the OSCommerce specific calls become deprecated then they must also be changed in the service.

Redevelopment solution

In the Redevelopment approach solution there is greater freedom from the system, as the developers can decide how the service is to be implemented, e.g. in the case of the Redevelopment service, a pure database call was extracted and implemented. The benefit is that the service code has lesser dependencies on the system, but at the same time requires the developer to code more, to achieve the same level of functionality and generality.

Comparison

The Redevelopment approach solution seems to have a greater level of independency of the system as opposed to the Hybrid approach solution, as it does not necessarily incorporate the existing code of the system. Furthermore the Hybrid approach solution typically limits the logic of the wrapper, to the development language which the system is implemented in, for simplicity, as opposed to the Redevelopment approach solution where any development language can be utilized for implementing the functionality.

8.5 Effort to achieve service principles

Hybrid solution:

The Hybrid approach solution service fulfils the SOA principles. It is reusable, as it is an independent entity which can be called by clients or other services as such. It exposes itself by a formal contract in the form of a WSDL document. It is loosely coupled as the service is independent of other services, and abstracts underlying logic by communicating via operations exposed by the WSDL. It's autonomous as the logic resides inside an explicit boundary, class and function, and is stateless

as state is not preserved between calls. Discoverability and composability has not been implemented in the case implementation, as the thesis limits itself from using UDDI.

To achieve the service principles addressed above, a function had to be defined and NuSOAP library included generating the WSDL document, the effort being seamlessly simple.

Redevelopment solution

The Redevelopment approach solution service fulfils the same SOA principles as the Hybrid approach solution.

Here a Java class and member function was implemented from which a WSDL document was generated. Also this approach required simple steps to achieve the principles.

Comparison

In both solutions it was possible to implement service which follows the SOA principles.

Chapter 9

9 Discussion

9.1 Results

The evaluation in chapter 8, in respects to important criteria for small business ecommerce found in chapter 5, shows that, there exist a lower complexity barrier in the development process of the Hybrid approach and in the level of required understanding of system source code. Nevertheless the Redevelopment approach promotes a higher level of extendibility, and lower dependency on the source system to be migrated. This will in the long term will ease the future development of the SOA system. Further the complexity level of development process will in the Redevelopment approach, decrease after the initial incremental implementation, as the initial setup steps will already have been undergone.

The easier extendibility and lower dependency level to the non-SOA system will enable a higher degree of flexibility, and even though a higher degree of code inspection and analysis must be preformed in the Redevelopment approach, it will result in a higher extent of understanding, of the services' functionality being developed.

Therefore when using a incremental strategy for migration to SOA, where feedback from small projects will be assed in order to decide further development, this report recommends, the use of the Redevelopment approach.

Furthermore, one thing that stood out was the uncomplicated process of implementing services that fulfilled the SOA principles (those that were in the scope of the thesis, not discoverability and composability).

This can be contributed to the technology available for Web service development, in this case, the NuSOAP and Ajax library which abstracted the level of complexity away, simplifying the development process, such as auto generating the shared formal contract in the form of the WSDL document and the abstraction of the SOAP communication. Therefore both services very quite simple to implement, even tough they still required some basic understanding of Web service technology.

9.2 Usage of the service

Some scenarios where the developed service could be used, is e.g. to expose the products on E-marketplaces that employ a SOA platform or as a service for collaboration partners to be able to service them self at any time, searching the businesses products. This could lead to new channels of sale, and improve relations with collaboration partners.

9.3 Is having a good approach enough

A good approach can act as guidance but in the end it is the actual situation of the individual company, its management, and the resources present that determines the outcome of any software implementation project.

9.4 Recommendation

The recommendation is stated to apply for small businesses using ecommerce, but is in no way restricted, and can just as well be employed by small development groups working in large organizations that e.g. are told to undergo a pilot project investigation of SOA, as they will probably have some of the same criteria as small businesses.

Chapter 10

10 Conclusion

The adoption of SOA by small businesses has not had the desired backing. This can to some extent be explained by small businesses not having the necessary level of resources that is required to venture into a new paradigm.

Furthermore it is not probable that non-SOA systems already present in small businesses will be replaced by new investments in SOA enabled system. It is more likely that future implementation projects carried out by these businesses, will involve the current systems present.

Migration where the system logic and/or data is moved to a new flexible environment, while retaining the original system's data and functionality, has therefore been the context of focus.

The thesis recognized a gap in literature dealing with, which service development approaches present, would be most suited for small businesses, specifically small ecommerce businesses, when migrating to SOA incrementally.

Therefore the overall problem statement presented in chapter 1, of determining which service development approach is best suited for small ecommerce businesses, following an incremental SOA migrating strategy, have been examined throughout the thesis.

Based on the outcome of the review of literature, on the subject of system integration in services to enable migration to SOA, three approaches were identified, from which two were determined to be feasible for further study by a case implementation and evaluation.

It can be concluded from the outcome of the case implementation evaluation that the Redevelopment approach is best suited for small ecommerce businesses, when migrating to SOA incrementally, as the approach offers higher extendibility, low dependency on the non-SOA source system, and a reduction in complexity as the incremental process progresses.

The thesis author is fully aware that, a single study of a case implementation alone, can not lead to explicit guidelines to be set up for small ecommerce businesses following an incremental SOA migration strategy, and therefore, it can be recommended that further studies should be conducted in order to get more insight in the area of SOA usage in small businesses.

Bibliography

- [1] Frank Gens. The IT Market's \$150B SMB "Long Tail", <http://blogs.idc.com/ie/?p=53>, 2006.
- [2] Frank Gens. TOP 10 PREDICTIONS, IDC Predictions 2007: Prospering in an Era of Hyper disruption, <http://cdn.idc.com/downloads/204631.pdf>, 2006.
- [3] Computer Economics. Service-Oriented Architecture: Adoption Trends in 2007, 2007.
- [4] Meriam-Webster. Meriam-Webster, <http://www.m-w.com/>, 2007.
- [5] Wikipedia. Legacy System, http://en.wikipedia.org/wiki/Legacy_system, 2007.
- [6] Wikipedia. Service-orientation, <http://en.wikipedia.org/wiki/Service-orientation>, 2007.
- [7] Thomas Erl. Revisiting the definitive SOA definition http://searchwebsites.techtarget.com/originalContent/0,289142,sid26_gci1044083,00.html, 2005.
- [8] Thomas Erl. Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall, 2005.
- [9] Google. Define:wSDL, <http://www.google.dk/search?hl=da&q=define%3Awsdl&btnG=S%C3%B8g&me>, 2007.
- [10] Google. Define:soap, http://www.google.dk/search?hl=da&defl=en&q=define:soap&sa=X&oi=glossary_definition&ct=title, 2007.
- [11] Google. Define:uddi, <http://www.google.dk/search?hl=da&q=define%3Auddi&btnG=S%C3%B8g&meta>, 2007.
- [12] Liam O'Brien, Len Bass, Paulo Merson. Quality Attributes and Service-Oriented Architectures, September 2005.
- [13] Jesús Bisbal, Deirdre Lawless, Bing Wu, and Jane Grimson. Legacy Information Systems: Issues and Directions, 1999.

Bibliography

- [14] Jesús Bisbal, Deirdre Lawless, Bing Wu, Jane Grimson. A Brief Review of Problems, Solutions and Research Issues, 1999.
- [15] Joe McKendrick. SOA business case: smaller is better, <http://blogs.zdnet.com/service-oriented/?p=903>, June 22, 2007.
- [16] Wikipedia, ECommerce: <http://en.wikipedia.org/wiki/E-commerce>, 2007.
- [17] Murray E. Jennex, Don Amoroso, Olayele Adelakun. E-commerce Infrastructure Success Factors for small Companies in Developing Economics, 2004.
- [18] Hanafi Mughrabi. A technical Investigation of SOA, and Ecommerce Systems, 2007
- [19] Joe McKendrick. SOA not for SMB, yet, <http://blogs.zdnet.com/service-oriented/?p=311>, June 7th. 2005.
- [20] Joe McKendrick. Small companies don't have time for SOA, <http://blogs.zdnet.com/service-oriented/?p=941>, 2007.
- [21] Archie Robostoff. Computerworld page 10, 2006.
- [22] Quocirca Ltd. Functional Re-use and SOA The Service Interface as an Enabler for Software Re-use Best Practice, 2007.
- [23] Zhuopeng Zhang, Hongji Yang. Incubating Services in Legacy Systems for Architectural Migration, <http://ieeexplore.ieee.org.globalproxy.cvt.dk/iel5/9444/29994/01371920.pdf>, 2004.
- [24] Gerardo Canfora, Anna Rita Fasolino, Gianni Frattolillo°, Porfirio Tramontana. Migrating Interactive Legacy Systems To Web Services, <http://ieeexplore.ieee.org.globalproxy.cvt.dk/iel5/10671/33675/01602355.pdf>, 2006.
- [25] Grace Lewis, Edwin Morris, Dennis Smith, Service-Oriented Migration and Reuse Technique (SMART), <http://ieeexplore.ieee.org.globalproxy.cvt.dk/iel5/11138/35653/01691651.pdf>, 2005.
- [26] W3C. Web Services Description Language (WSDL) 1.1, <http://www.w3.org/TR/wsdl>, 2001.
- [27] W3C. Web Simple Object Access Protocol (SOAP) 1.1, <http://www.w3.org/>

Bibliography

TR/2000/NOTE-SOAP-20000508/, 2000.

[28] Programming with NuSOAP Using WSDL, http://www.scottnichol.com/nusoap_progwsdl.htm, 2003.

[29] O'Reilly. Tomcat def, <http://www.onjava.com/pub/ct/33>, 2000.

[30] Wikipedia. Apache Axis, [en.wikipedia.org/wiki/Axis_\(computer_program\)](http://en.wikipedia.org/wiki/Axis_(computer_program)), 2007.

Appendix A - Setup

Setup steps for OSCommerce:

1. JSAS 1.10 was downloaded <http://joomlancode.org/gf/project/jsas/frs/>
2. OSCommerce 2.2.1 was downloaded <http://www.oscommerce.com/solutions/downloads>
3. OSCommerce zipfile was renamed to joomla.zip and replaced the preexisting joomla.zip zipfile in JSAS.
4. JSAS was installed together with OSCommerce on virtual drive W:
5. OSCommerce were setup.
6. OSCommerce main folder holding al functionality is located at the following path W:../oscommerce-2.2rc1/catalog/

Setup and test steps for Hybrid approach:

1. Prerequisite is that setup steps for OSCommerce are executed first.
2. NuSOAP 0.7.2 PHP Web service library was downloaded http://sourceforge.net/project/showfiles.php?group_id=57663
3. NuSOAP was unzipped and placed at the folder W:../oscommerce 2.2rc1/catalog/lib for easy reference purpose.
4. productSearch.php was developed and placed in W:../oscommerce-2.2rc1/catalog/
5. WSDL was auto generated by going to W:../oscommerce-2.2rc1/catalog/productSearch.php?wsdl

6. productSearchClient.php was developed and placed in W:../oscommerce-2.2rc1/catalog/

7. Service was tested by hard coding input parameter in client and going to

```
// Call the SOAP method  
$result = $client->call('prodSearch', array('search' =>  
'beloved'));
```

8. W:../oscommerce-2.2rc1/catalog/productSearchClient.php

Setup and test steps for Redevelopment approach:

1. Prerequisite is that setup steps for OSCommerce are executed first, and that java CLASSPATH and JAVAHOME is set correct.
2. Jakarta-Tomcat 4.0.1 Web server had to be downloaded <http://jakarta.apache.org/tomcat/index.html> and installed to folder ../jakarta-tomcat-4.0.1
3. Apache Axis 1.0 java Web service framework had to be downloaded <http://xml.apache.org/axis/index.html>, unzipped to ../axis-1_0, copied and placed at ../jakarta-tomcat-4.0.1/webapps/axis
4. Xerces 1.4.4 had to be downloaded <http://xml.apache.org/xerces2-j/index.html> and xerces.jar had to be placed in ../jakarta-tomcat-4.0.1/webapps/axis/WEB-INF/lib
5. <Context path="/axis" docBase="axis" debug="0" reloadable="true"/> is added to server.xml file in the directory jakarta-tomcat-4.0.1/conf
6. Set CLASSPATH to: ../axis-1_0/lib/axis.jar;../axis-1_0/lib/log4j-core.jar;../xerces-1_4_4/xerces.jar;../axis-1_0/lib/wsdl4j.jar;../axis-1_0/lib/clutil.jar;../axis-1_0/lib/commons-logging.jar;../axis-1_0/lib/commons-discovery.jar;../axis-1_0/lib/saaj.jar;../axis-1_0/lib/jaxrpc.jar;../Java/jdk1.5.0_08/bin;../Java/jdk1.5.0_08/lib/tools.jar;../jakarta-tomcat-4.0.1/jakarta-tomcat-

4.0.1/webapps/axis/localhost;../Java/jdk1.5.0_08/jre/lib/ext/mysql-connector-java-3.0.17-ga-bin.jar

7. Execute in commando prompt: `java org.apache.axis.wsdl.WSDL2Java -p http://localhost:8080/axis/productSearch.jws?wsdl` to generate Java classes from the Web service's WSDL description, which will be used for test client implementation.
8. `javac productSearchClient.java`
9. `java localhost.productSearchClient DVD`

For more detail information the reader is advised to go to tutorial on <http://www.javaworld.com/javaworld/jw-01-2002/jw-0125-axis.html>

Appendix B - Product search functionality

advanced_search_result.php:

```
<?php
/*
   $Id: advanced_search_result.php,v 1.72 2003/06/23 06:50:11
   project3000 Exp $

   osCommerce, Open Source E-Commerce Solutions
   http://www.oscommerce.com

   Copyright (c) 2003 osCommerce

   Released under the GNU General Public License
*/

require('includes/application_top.php');

require(DIR_WS_LANGUAGES . $language . '/' .
FILENAME_ADVANCED_SEARCH);

$error = false;

if ( (isset($HTTP_GET_VARS['keywords']) &&
empty($HTTP_GET_VARS['keywords'])) &&
    (isset($HTTP_GET_VARS['dfrom']) &&
(empty($HTTP_GET_VARS['dfrom']) || ($HTTP_GET_VARS['dfrom']
== DOB_FORMAT_STRING))) &&
    (isset($HTTP_GET_VARS['dto']) &&
(empty($HTTP_GET_VARS['dto']) || ($HTTP_GET_VARS['dto'] ==
DOB_FORMAT_STRING))) &&
    (isset($HTTP_GET_VARS['pfrom']) &&
!is_numeric($HTTP_GET_VARS['pfrom'])) &&
    (isset($HTTP_GET_VARS['pto']) &&
!is_numeric($HTTP_GET_VARS['pto'])) ) {
    $error = true;

    $messageStack->add_session('search',
ERROR_AT_LEAST_ONE_INPUT);
} else {
    $dfrom = '';
    $dto = '';
    $pfrom = '';
    $pto = '';
    $keywords = '';

    if (isset($HTTP_GET_VARS['dfrom'])) {
        $dfrom = (($HTTP_GET_VARS['dfrom'] ==
```

Appendix B – Product search functionality

```
DOB_FORMAT_STRING) ? '' : $HTTP_GET_VARS['dfrom'];
}

    if (isset($HTTP_GET_VARS['dto'])) {
        $dto = (($HTTP_GET_VARS['dto'] == DOB_FORMAT_STRING) ?
'' : $HTTP_GET_VARS['dto']);
    }

    if (isset($HTTP_GET_VARS['pfrom'])) {
        $pfrom = $HTTP_GET_VARS['pfrom'];
    }

    if (isset($HTTP_GET_VARS['pto'])) {
        $pto = $HTTP_GET_VARS['pto'];
    }

    if (isset($HTTP_GET_VARS['keywords'])) {
        $keywords = $HTTP_GET_VARS['keywords'];
    }

    $date_check_error = false;
    if (tep_not_null($dfrom)) {
        if (!tep_checkdate($dfrom, DOB_FORMAT_STRING,
$dfrom_array)) {
            $error = true;
            $date_check_error = true;

            $messageStack->add_session('search',
ERROR_INVALID_FROM_DATE);
        }
    }

    if (tep_not_null($dto)) {
        if (!tep_checkdate($dto, DOB_FORMAT_STRING,
$dto_array)) {
            $error = true;
            $date_check_error = true;

            $messageStack->add_session('search',
ERROR_INVALID_TO_DATE);
        }
    }

    if (($date_check_error == false) && tep_not_null($dfrom)
&& tep_not_null($dto)) {
        if (mktime(0, 0, 0, $dfrom_array[1], $dfrom_array[2],
$dfrom_array[0]) > mktime(0, 0, 0, $dto_array[1],
$dto_array[2], $dto_array[0])) {
            $error = true;

            $messageStack->add_session('search',
ERROR_TO_DATE_LESS_THAN_FROM_DATE);
        }
    }

    $price_check_error = false;
```

```

if (tep_not_null($pfrom)) {
    if (!settype($pfrom, 'double')) {
        $error = true;
        $price_check_error = true;

        $messageStack->add_session('search',
ERROR_PRICE_FROM_MUST_BE_NUM);
    }
}

if (tep_not_null($pto)) {
    if (!settype($pto, 'double')) {
        $error = true;
        $price_check_error = true;

        $messageStack->add_session('search',
ERROR_PRICE_TO_MUST_BE_NUM);
    }
}

if (($price_check_error == false) && is_float($pfrom) &&
is_float($pto)) {
    if ($pfrom >= $pto) {
        $error = true;

        $messageStack->add_session('search',
ERROR_PRICE_TO_LESS_THAN_PRICE_FROM);
    }
}

if (tep_not_null($keywords)) {
    if (!tep_parse_search_string($keywords,
$search_keywords)) {
        $error = true;

        $messageStack->add_session('search',
ERROR_INVALID_KEYWORDS);
    }
}

if (empty($dfrom) && empty($dto) && empty($pfrom) &&
empty($pto) && empty($keywords)) {
    $error = true;

    $messageStack->add_session('search',
ERROR_AT_LEAST_ONE_INPUT);
}

if ($error == true) {
    tep_redirect(tep_href_link(FILENAME_ADVANCED_SEARCH,
tep_get_all_get_params(), 'NONSSL', true, false));
}

$breadcrumb->add(NAVBAR_TITLE_1,
tep_href_link(FILENAME_ADVANCED_SEARCH));

```



```

    $breadcrumb->add(NAVBAR_TITLE_2,
    tep_href_link(FILENAME_ADVANCED_SEARCH_RESULT,
    tep_get_all_get_params(), 'NONSSL', true, false));
?>
<!doctype html public "-//W3C//DTD HTML 4.01
Transitional//EN">
<html <?php echo HTML_PARAMS; ?>>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=<?php echo CHARSET; ?>">
<base href="<?php echo (($request_type == 'SSL') ?
HTTPS_SERVER : HTTP_SERVER) . DIR_WS_CATALOG; ?>">
<title><?php echo TITLE; ?></title>
<link rel="stylesheet" type="text/css"
href="stylesheet.css">
</head>
<body marginwidth="0" marginheight="0" topmargin="0"
bottommargin="0" leftmargin="0" rightmargin="0">
<!-- header //-->
<?php require(DIR_WS_INCLUDES . 'header.php'); ?>
<!-- header_eof //-->

<!-- body //-->
<table border="0" width="100%" cellpadding="3"
cellpadding="3">
    <tr>
        <td width="<?php echo BOX_WIDTH; ?>" valign="top"><table
border="0" width="<?php echo BOX_WIDTH; ?>" cellpadding="0"
cellpadding="2">
<!-- left_navigation //-->
<?php require(DIR_WS_INCLUDES . 'column_left.php'); ?>
<!-- left_navigation_eof //-->
</table></td>
<!-- body_text //-->
        <td width="100%" valign="top"><table border="0"
width="100%" cellpadding="0" cellspacing="0">
            <tr>
                <td><table border="0" width="100%" cellpadding="0">
                    <tr>
                        <td class="pageHeading"><?php echo
HEADING_TITLE_2; ?></td>
                        <td class="pageHeading" align="right"><?php echo
tep_image(DIR_WS_IMAGES . 'table_background_browse.gif',
HEADING_TITLE_2, HEADING_IMAGE_WIDTH, HEADING_IMAGE_HEIGHT);
?></td>
                    </tr>
                </table></td>
            </tr>
            <tr>
                <td><?php echo tep_draw_separator('pixel_trans.gif',
'100%', '10'); ?></td>
            </tr>
            <tr>
                <td>
<?php

```

Appendix B – Product search functionality

```
// create column list
$define_list = array('PRODUCT_LIST_MODEL' =>
PRODUCT_LIST_MODEL,
                    'PRODUCT_LIST_NAME' =>
PRODUCT_LIST_NAME,
                    'PRODUCT_LIST_MANUFACTURER' =>
PRODUCT_LIST_MANUFACTURER,
                    'PRODUCT_LIST_PRICE' =>
PRODUCT_LIST_PRICE,
                    'PRODUCT_LIST_QUANTITY' =>
PRODUCT_LIST_QUANTITY,
                    'PRODUCT_LIST_WEIGHT' =>
PRODUCT_LIST_WEIGHT,
                    'PRODUCT_LIST_IMAGE' =>
PRODUCT_LIST_IMAGE,
                    'PRODUCT_LIST_BUY_NOW' =>
PRODUCT_LIST_BUY_NOW);

asort($define_list);

$column_list = array();
reset($define_list);
while (list($key, $value) = each($define_list)) {
    if ($value > 0) $column_list[] = $key;
}

$select_column_list = '';

for ($i=0, $n=sizeof($column_list); $i<$n; $i++) {
    switch ($column_list[$i]) {
        case 'PRODUCT_LIST_MODEL':
            $select_column_list .= 'p.products_model, ';
            break;
        case 'PRODUCT_LIST_MANUFACTURER':
            $select_column_list .= 'm.manufacturers_name, ';
            break;
        case 'PRODUCT_LIST_QUANTITY':
            $select_column_list .= 'p.products_quantity, ';
            break;
        case 'PRODUCT_LIST_IMAGE':
            $select_column_list .= 'p.products_image, ';
            break;
        case 'PRODUCT_LIST_WEIGHT':
            $select_column_list .= 'p.products_weight, ';
            break;
    }
}

$select_str = "select distinct " . $select_column_list . "
m.manufacturers_id, p.products_id, pd.products_name,
p.products_price, p.products_tax_class_id, IF(s.status,
s.specials_new_products_price, NULL) as
specials_new_products_price, IF(s.status,
s.specials_new_products_price, p.products_price) as
final_price ";
```

Appendix B – Product search functionality

```
if ( (DISPLAY_PRICE_WITH_TAX == 'true') &&
(tep_not_null($pfrom) || tep_not_null($pto)) ) {
    $select_str .= ", SUM(tr.tax_rate) as tax_rate ";
}

$from_str = "from " . TABLE_PRODUCTS . " p left join " .
TABLE_MANUFACTURERS . " m using(manufacturers_id) left join
" . TABLE_SPECIALS . " s on p.products_id = s.products_id";

if ( (DISPLAY_PRICE_WITH_TAX == 'true') &&
(tep_not_null($pfrom) || tep_not_null($pto)) ) {
    if (!tep_session_is_registered('customer_country_id')) {
        $customer_country_id = STORE_COUNTRY;
        $customer_zone_id = STORE_ZONE;
    }
    $from_str .= " left join " . TABLE_TAX_RATES . " tr on
p.products_tax_class_id = tr.tax_class_id left join " .
TABLE_ZONES_TO_GEO_ZONES . " gz on tr.tax_zone_id =
gz.geo_zone_id and (gz.zone_country_id is null or
gz.zone_country_id = '0' or gz.zone_country_id = '" .
(int)$customer_country_id . "') and (gz.zone_id is null or
gz.zone_id = '0' or gz.zone_id = '" . (int)$customer_zone_id
. "')";
}

$from_str .= ", " . TABLE_PRODUCTS_DESCRIPTION . " pd, " .
TABLE_CATEGORIES . " c, " . TABLE_PRODUCTS_TO_CATEGORIES . "
p2c";

$where_str = " where p.products_status = '1' and
p.products_id = pd.products_id and pd.language_id = '" .
(int)$languages_id . "' and p.products_id = p2c.products_id
and p2c.categories_id = c.categories_id ";

if (isset($HTTP_GET_VARS['categories_id']) &&
tep_not_null($HTTP_GET_VARS['categories_id'])) {
    if (isset($HTTP_GET_VARS['inc_subcat']) &&
($HTTP_GET_VARS['inc_subcat'] == '1')) {
        $subcategories_array = array();
        tep_get_subcategories($subcategories_array,
$HTTP_GET_VARS['categories_id']);

        $where_str .= " and p2c.products_id = p.products_id
and p2c.products_id = pd.products_id and (p2c.categories_id
= '" . (int)$HTTP_GET_VARS['categories_id'] . "'";

        for ($i=0, $n=sizeof($subcategories_array); $i<$n;
$i++) {
            $where_str .= " or p2c.categories_id = '" .
(int)$subcategories_array[$i] . "'";
        }

        $where_str .= ")";
    } else {
        $where_str .= " and p2c.products_id = p.products_id
and p2c.products_id = pd.products_id and pd.language_id = '"
```

```

. (int)$languages_id . "' and p2c.categories_id = '" .
(int)$HTTP_GET_VARS['categories_id'] . "'";
    }
}

    if (isset($HTTP_GET_VARS['manufacturers_id']) &&
tep_not_null($HTTP_GET_VARS['manufacturers_id'])) {
        $where_str .= " and m.manufacturers_id = '" .
(int)$HTTP_GET_VARS['manufacturers_id'] . "'";
    }

    if (isset($search_keywords) && (sizeof($search_keywords) >
0)) {
        $where_str .= " and (";
        for ($i=0, $n=sizeof($search_keywords); $i<$n; $i++ ) {
            switch ($search_keywords[$i]) {
                case '(':
                case ')':
                case 'and':
                case 'or':
                    $where_str .= " " . $search_keywords[$i] . " ";
                    break;
                default:
                    $keyword =
tep_db_prepare_input($search_keywords[$i]);
                    $where_str .= "(pd.products_name like '%" .
tep_db_input($keyword) . "%' or p.products_model like '%" .
tep_db_input($keyword) . "%' or m.manufacturers_name like
 '%" . tep_db_input($keyword) . "%'";
                    if (isset($HTTP_GET_VARS['search_in_description'])
&& ($HTTP_GET_VARS['search_in_description'] == '1'))
                    $where_str .= " or pd.products_description like '%" .
tep_db_input($keyword) . "%'";
                    $where_str .= ')';
                    break;
            }
        }
        $where_str .= " )";
    }

    if (tep_not_null($dfrom)) {
        $where_str .= " and p.products_date_added >= '" .
tep_date_raw($dfrom) . "'";
    }

    if (tep_not_null($dto)) {
        $where_str .= " and p.products_date_added <= '" .
tep_date_raw($dto) . "'";
    }

    if (tep_not_null($pfrom)) {
        if ($currencies->is_set($currency)) {
            $rate = $currencies->get_value($currency);

            $pfrom = $pfrom / $rate;
        }
    }

```

```

}

if (tep_not_null($pto)) {
    if (isset($rate)) {
        $pto = $pto / $rate;
    }
}

if (DISPLAY_PRICE_WITH_TAX == 'true') {
    if ($pfrom > 0) $where_str .= " and (IF(s.status,
s.specials_new_products_price, p.products_price) *
if(gz.geo_zone_id is null, 1, 1 + (tr.tax_rate / 100) ) >= "
. (double)$pfrom . ")";
    if ($pto > 0) $where_str .= " and (IF(s.status,
s.specials_new_products_price, p.products_price) *
if(gz.geo_zone_id is null, 1, 1 + (tr.tax_rate / 100) ) <= "
. (double)$pto . ")";
    } else {
        if ($pfrom > 0) $where_str .= " and (IF(s.status,
s.specials_new_products_price, p.products_price) >= "
. (double)$pfrom . ")";
        if ($pto > 0) $where_str .= " and (IF(s.status,
s.specials_new_products_price, p.products_price) <= "
. (double)$pto . ")";
    }

    if ( (DISPLAY_PRICE_WITH_TAX == 'true') &&
(tep_not_null($pfrom) || tep_not_null($pto)) ) {
        $where_str .= " group by p.products_id,
tr.tax_priority";
    }

    if ( (!isset($HTTP_GET_VARS['sort'])) || (!ereg('[1-
8][ad]', $HTTP_GET_VARS['sort'])) ||
(substr($HTTP_GET_VARS['sort'], 0, 1) >
sizeof($column_list)) ) {
        for ($i=0, $n=sizeof($column_list); $i<$n; $i++) {
            if ($column_list[$i] == 'PRODUCT_LIST_NAME') {
                $HTTP_GET_VARS['sort'] = $i+1 . 'a';
                $order_str = ' order by pd.products_name';
                break;
            }
        }
    } else {
        $sort_col = substr($HTTP_GET_VARS['sort'], 0, 1);
        $sort_order = substr($HTTP_GET_VARS['sort'], 1);
        $order_str = ' order by ';
        switch ($column_list[$sort_col-1]) {
            case 'PRODUCT_LIST_MODEL':
                $order_str .= "p.products_model " . ($sort_order ==
'd' ? "desc" : "") . ", pd.products_name";
                break;
            case 'PRODUCT_LIST_NAME':
                $order_str .= "pd.products_name " . ($sort_order ==
'd' ? "desc" : "");
                break;
        }
    }
}

```

```

        case 'PRODUCT_LIST_MANUFACTURER':
            $order_str .= "m.manufacturers_name " . ($sort_order
== 'd' ? "desc" : "") . ", pd.products_name";
            break;
        case 'PRODUCT_LIST_QUANTITY':
            $order_str .= "p.products_quantity " . ($sort_order
== 'd' ? "desc" : "") . ", pd.products_name";
            break;
        case 'PRODUCT_LIST_IMAGE':
            $order_str .= "pd.products_name";
            break;
        case 'PRODUCT_LIST_WEIGHT':
            $order_str .= "p.products_weight " . ($sort_order ==
'd' ? "desc" : "") . ", pd.products_name";
            break;
        case 'PRODUCT_LIST_PRICE':
            $order_str .= "final_price " . ($sort_order == 'd' ?
"desc" : "") . ", pd.products_name";
            break;
    }
}

$listing_sql = $select_str . $from_str . $where_str .
$order_str;

require(DIR_WS_MODULES . FILENAME_PRODUCT_LISTING);
?>
    </td>
</tr>
<tr>
    <td><?php echo tep_draw_separator('pixel_trans.gif',
'100%', '10'); ?></td>
</tr>
<tr>
    <td class="main"><?php echo '<a href="' .
tep_href_link(FILENAME_ADVANCED_SEARCH,
tep_get_all_get_params(array('sort', 'page')), 'NONSSL',
true, false) . '>' . tep_image_button('button_back.gif',
IMAGE_BUTTON_BACK) . '</a>'; ?></td>
</tr>
</table></td>
<!-- body_text_eof //-->
    <td width="<?php echo BOX_WIDTH; ?>" valign="top"><table
border="0" width="<?php echo BOX_WIDTH; ?>" cellpadding="0"
cellspacing="2">
<!-- right_navigation //-->
<?php require(DIR_WS_INCLUDES . 'column_right.php'); ?>
<!-- right_navigation_eof //-->
    </table></td>
</tr>
</table>
<!-- body_eof //-->

<!-- footer //-->
<?php require(DIR_WS_INCLUDES . 'footer.php'); ?>
<!-- footer_eof //-->

```

```
<br>  
</body>  
</html>  
<?php require(DIR_WS_INCLUDES . 'application_bottom.php');  

```

Appendix C – Service implementation

Hybrid approach service:

productSearch.php

```
<?php
//require('includes/functions/general.php');
//require('includes/functions/database.php');
//require('includes/classes/split_page_results.php');
//require('includes/application_top.php');
// require(DIR_WS_LANGUAGES . $language . '/' .
FILENAME_ADVANCED_SEARCH);

// Pull in the NuSOAP code
require_once('lib/nusoap.php');
// Create the server instance
$server = new soap_server();
// Initialize WSDL support
$server->configureWSDL('productSearch',
'urn:productSearch');
// Register the method to expose
$server->register('prodSearch',
// method name
    array('search' => 'xsd:string'),
// input parameters
    array('return' => 'xsd:string'),
// output parameters
    'urn:productSearch',
// namespace
    'urn:productSearch#prodSearch',
// soapaction
    'rpc',
// style
    'encoded',
// use
    'Searches OSCommerce product catalog and returns result'
        // documentation
);
// Define the method as a PHP function
function prodSearch($search) {

    require('includes/application_top.php');
    require(DIR_WS_LANGUAGES . $language . '/' .
```


Appendix C – Service implementation

```
FILENAME_ADVANCED_SEARCH);

// create column list

/*
   $Id: advanced_search_result.php,v 1.72 2003/06/23 06:50:11
project3000 Exp $

   osCommerce, Open Source E-Commerce Solutions
   http://www.oscommerce.com

   Copyright (c) 2003 osCommerce

   Released under the GNU General Public License
*/

$error = false;

if ( (isset($HTTP_GET_VARS['keywords']) &&
empty($HTTP_GET_VARS['keywords'])) &&
    (isset($HTTP_GET_VARS['dfrom']) &&
(empty($HTTP_GET_VARS['dfrom']) || ($HTTP_GET_VARS['dfrom']
== DOB_FORMAT_STRING))) &&
    (isset($HTTP_GET_VARS['dto']) &&
(empty($HTTP_GET_VARS['dto']) || ($HTTP_GET_VARS['dto'] ==
DOB_FORMAT_STRING))) &&
    (isset($HTTP_GET_VARS['pfrom']) &&
!is_numeric($HTTP_GET_VARS['pfrom'])) &&
    (isset($HTTP_GET_VARS['pto']) &&
!is_numeric($HTTP_GET_VARS['pto'])) ) {
    $error = true;

    $messageStack->add_session('search',
ERROR_AT_LEAST_ONE_INPUT);
} else {
    $dfrom = '';
    $dto = '';
    $pfrom = '';
    $pto = '';
    $keywords = '';

    if (isset($HTTP_GET_VARS['dfrom'])) {
        $dfrom = (($HTTP_GET_VARS['dfrom'] ==
DOB_FORMAT_STRING) ? '' : $HTTP_GET_VARS['dfrom']);
    }

    if (isset($HTTP_GET_VARS['dto'])) {
        $dto = (($HTTP_GET_VARS['dto'] == DOB_FORMAT_STRING) ?
'' : $HTTP_GET_VARS['dto']);
    }

    if (isset($HTTP_GET_VARS['pfrom'])) {
        $pfrom = $HTTP_GET_VARS['pfrom'];
    }

    if (isset($HTTP_GET_VARS['pto'])) {
```

```

    $pto = $HTTP_GET_VARS['pto'];
}

if (isset($HTTP_GET_VARS['keywords'])) {
    $keywords = $HTTP_GET_VARS['keywords'];
}

// HMU ADDED
// sets the value of keywords to my search criteria
$keywords = $search;

$date_check_error = false;
if (tep_not_null($dfrom)) {
    if (!tep_checkdate($dfrom, DOB_FORMAT_STRING,
$dfrom_array)) {
        $error = true;
        $date_check_error = true;

//      $messageStack->add_session('search',
ERROR_INVALID_FROM_DATE);
    }
}

if (tep_not_null($dto)) {
    if (!tep_checkdate($dto, DOB_FORMAT_STRING,
$dto_array)) {
        $error = true;
        $date_check_error = true;

//      $messageStack->add_session('search',
ERROR_INVALID_TO_DATE);
    }
}

if (($date_check_error == false) && tep_not_null($dfrom)
&& tep_not_null($dto)) {
    if (mktime(0, 0, 0, $dfrom_array[1], $dfrom_array[2],
$dfrom_array[0]) > mktime(0, 0, 0, $dto_array[1],
$dto_array[2], $dto_array[0])) {
        $error = true;

//      $messageStack->add_session('search',
ERROR_TO_DATE_LESS_THAN_FROM_DATE);
    }
}

$price_check_error = false;
if (tep_not_null($pfrom)) {
    if (!settype($pfrom, 'double')) {
        $error = true;
        $price_check_error = true;

//      $messageStack->add_session('search',
ERROR_PRICE_FROM_MUST_BE_NUM);
    }
}

```

```

    }

    if (tep_not_null($pto)) {
        if (!settype($pto, 'double')) {
            $error = true;
            $price_check_error = true;

            // $messageStack->add_session('search',
            ERROR_PRICE_TO_MUST_BE_NUM);
        }
    }

    if (($price_check_error == false) && is_float($pfrom) &&
    is_float($pto)) {
        if ($pfrom >= $pto) {
            $error = true;

            // $messageStack->add_session('search',
            ERROR_PRICE_TO_LESS_THAN_PRICE_FROM);
        }
    }

    if (tep_not_null($keywords)) {
        if (!tep_parse_search_string($keywords,
        $search_keywords)) {
            $error = true;

            // $messageStack->add_session('search',
            ERROR_INVALID_KEYWORDS);
        }
    }
}

if (empty($dfrom) && empty($dto) && empty($pfrom) &&
empty($pto) && empty($keywords)) {
    $error = true;

    // $messageStack->add_session('search',
    ERROR_AT_LEAST_ONE_INPUT);
}

// if ($error == true) {
//     tep_redirect(tep_href_link(FILENAME_ADVANCED_SEARCH,
//     tep_get_all_get_params(), 'NONSSL', true, false));
// }

// $breadcrumb->add(NAVBAR_TITLE_1,
// tep_href_link(FILENAME_ADVANCED_SEARCH));
// $breadcrumb->add(NAVBAR_TITLE_2,
// tep_href_link(FILENAME_ADVANCED_SEARCH_RESULT,
// tep_get_all_get_params(), 'NONSSL', true, false));

$define_list = array('PRODUCT_LIST_MODEL' =>
PRODUCT_LIST_MODEL,
                    'PRODUCT_LIST_NAME' =>
PRODUCT_LIST_NAME,

```

Appendix C – Service implementation

```
                'PRODUCT_LIST_MANUFACTURER' =>
PRODUCT_LIST_MANUFACTURER,
                'PRODUCT_LIST_PRICE' =>
PRODUCT_LIST_PRICE,
                'PRODUCT_LIST_QUANTITY' =>
PRODUCT_LIST_QUANTITY,
                'PRODUCT_LIST_WEIGHT' =>
PRODUCT_LIST_WEIGHT,
                'PRODUCT_LIST_IMAGE' =>
PRODUCT_LIST_IMAGE,
                'PRODUCT_LIST_BUY_NOW' =>
PRODUCT_LIST_BUY_NOW);

    asort($define_list);

    $column_list = array();
    reset($define_list);
    while (list($key, $value) = each($define_list)) {
        if ($value > 0) $column_list[] = $key;
    }

    $select_column_list = '';

    for ($i=0, $n=sizeof($column_list); $i<$n; $i++) {
        switch ($column_list[$i]) {
            case 'PRODUCT_LIST_MODEL':
                $select_column_list .= 'p.products_model, ';
                break;
            case 'PRODUCT_LIST_MANUFACTURER':
                $select_column_list .= 'm.manufacturers_name, ';
                break;
            case 'PRODUCT_LIST_QUANTITY':
                $select_column_list .= 'p.products_quantity, ';
                break;
            case 'PRODUCT_LIST_IMAGE':
                $select_column_list .= 'p.products_image, ';
                break;
            case 'PRODUCT_LIST_WEIGHT':
                $select_column_list .= 'p.products_weight, ';
                break;
        }
    }

    $select_str = "select distinct " . $select_column_list . "
m.manufacturers_id, p.products_id, pd.products_name,
p.products_price, p.products_tax_class_id, IF(s.status,
s.specials_new_products_price, NULL) as
specials_new_products_price, IF(s.status,
s.specials_new_products_price, p.products_price) as
final_price ";

    if ( (DISPLAY_PRICE_WITH_TAX == 'true') &&
(tep_not_null($pfrom) || tep_not_null($pto)) ) {
        $select_str .= ", SUM(tr.tax_rate) as tax_rate ";
    }
}
```

Appendix C – Service implementation

```
$from_str = "from " . TABLE_PRODUCTS . " p left join " .
TABLE_MANUFACTURERS . " m using(manufacturers_id) left join
" . TABLE_SPECIALS . " s on p.products_id = s.products_id";

if ( (DISPLAY_PRICE_WITH_TAX == 'true') &&
(tep_not_null($pfrom) || tep_not_null($pto)) ) {
    if (!tep_session_is_registered('customer_country_id')) {
        $customer_country_id = STORE_COUNTRY;
        $customer_zone_id = STORE_ZONE;
    }
    $from_str .= " left join " . TABLE_TAX_RATES . " tr on
p.products_tax_class_id = tr.tax_class_id left join " .
TABLE_ZONES_TO_GEO_ZONES . " gz on tr.tax_zone_id =
gz.geo_zone_id and (gz.zone_country_id is null or
gz.zone_country_id = '0' or gz.zone_country_id = '" .
(int)$customer_country_id . "'" ) and (gz.zone_id is null or
gz.zone_id = '0' or gz.zone_id = '" . (int)$customer_zone_id
. "'";
}

$from_str .= ", " . TABLE_PRODUCTS_DESCRIPTION . " pd, " .
TABLE_CATEGORIES . " c, " . TABLE_PRODUCTS_TO_CATEGORIES . "
p2c";

$where_str = " where p.products_status = '1' and
p.products_id = pd.products_id and pd.language_id = '" .
(int)$languages_id . "'" and p.products_id = p2c.products_id
and p2c.categories_id = c.categories_id ";

if (isset($HTTP_GET_VARS['categories_id']) &&
tep_not_null($HTTP_GET_VARS['categories_id'])) {
    if (isset($HTTP_GET_VARS['inc_subcat']) &&
($HTTP_GET_VARS['inc_subcat'] == '1')) {
        $subcategories_array = array();
        tep_get_subcategories($subcategories_array,
$HTTP_GET_VARS['categories_id']);

        $where_str .= " and p2c.products_id = p.products_id
and p2c.products_id = pd.products_id and (p2c.categories_id
= '" . (int)$HTTP_GET_VARS['categories_id'] . "'";

        for ($i=0, $n=sizeof($subcategories_array); $i<$n;
$i++) {
            $where_str .= " or p2c.categories_id = '" .
(int)$subcategories_array[$i] . "'";
        }

        $where_str .= ")";
    } else {
        $where_str .= " and p2c.products_id = p.products_id
and p2c.products_id = pd.products_id and pd.language_id = '"
. (int)$languages_id . "'" and p2c.categories_id = '" .
(int)$HTTP_GET_VARS['categories_id'] . "'";
    }
}
}
```

```

    if (isset($HTTP_GET_VARS['manufacturers_id']) &&
tep_not_null($HTTP_GET_VARS['manufacturers_id'])) {
        $where_str .= " and m.manufacturers_id = " .
(int)$HTTP_GET_VARS['manufacturers_id'] . "'";
    }

    if (isset($search_keywords) && (sizeof($search_keywords) >
0)) {
        $where_str .= " and (";
        for ($i=0, $n=sizeof($search_keywords); $i<$n; $i++) {
            switch ($search_keywords[$i]) {
                case '(':
                case ')':
                case 'and':
                case 'or':
                    $where_str .= " " . $search_keywords[$i] . " ";
                    break;
                default:
                    $keyword =
tep_db_prepare_input($search_keywords[$i]);
                    $where_str .= "(pd.products_name like '%" .
tep_db_input($keyword) . "%' or p.products_model like '%" .
tep_db_input($keyword) . "%' or m.manufacturers_name like
 '%" . tep_db_input($keyword) . "%'";
                    if (isset($HTTP_GET_VARS['search_in_description'])
&& ($HTTP_GET_VARS['search_in_description'] == '1'))
                    $where_str .= " or pd.products_description like '%" .
tep_db_input($keyword) . "%'";
                    $where_str .= ')';
                    break;
            }
        }
        $where_str .= " )";
    }

    if (tep_not_null($dfrom)) {
        $where_str .= " and p.products_date_added >= " .
tep_date_raw($dfrom) . "'";
    }

    if (tep_not_null($dto)) {
        $where_str .= " and p.products_date_added <= " .
tep_date_raw($dto) . "'";
    }

    if (tep_not_null($pfrom)) {
        if ($currencies->is_set($currency)) {
            $rate = $currencies->get_value($currency);

            $pfrom = $pfrom / $rate;
        }
    }

    if (tep_not_null($pto)) {
        if (isset($rate)) {
            $pto = $pto / $rate;
        }
    }

```

```

    }
}

if (DISPLAY_PRICE_WITH_TAX == 'true') {
    if ($pfrom > 0) $where_str .= " and (IF(s.status,
s.specials_new_products_price, p.products_price) *
if(gz.geo_zone_id is null, 1, 1 + (tr.tax_rate / 100) ) >= "
. (double)$pfrom . ")";
    if ($pto > 0) $where_str .= " and (IF(s.status,
s.specials_new_products_price, p.products_price) *
if(gz.geo_zone_id is null, 1, 1 + (tr.tax_rate / 100) ) <= "
. (double)$pto . ")";
} else {
    if ($pfrom > 0) $where_str .= " and (IF(s.status,
s.specials_new_products_price, p.products_price) >= "
. (double)$pfrom . ")";
    if ($pto > 0) $where_str .= " and (IF(s.status,
s.specials_new_products_price, p.products_price) <= "
. (double)$pto . ")";
}

if ( (DISPLAY_PRICE_WITH_TAX == 'true') &&
(tep_not_null($pfrom) || tep_not_null($pto)) ) {
    $where_str .= " group by p.products_id,
tr.tax_priority";
}

if ( (!isset($HTTP_GET_VARS['sort'])) || (!ereg('[1-
8][ad]', $HTTP_GET_VARS['sort'])) ||
(substr($HTTP_GET_VARS['sort'], 0, 1) >
sizeof($column_list)) ) {
    for ($i=0, $n=sizeof($column_list); $i<$n; $i++) {
        if ($column_list[$i] == 'PRODUCT_LIST_NAME') {
            $HTTP_GET_VARS['sort'] = $i+1 . 'a';
            $order_str = ' order by pd.products_name';
            break;
        }
    }
} else {
    $sort_col = substr($HTTP_GET_VARS['sort'], 0 , 1);
    $sort_order = substr($HTTP_GET_VARS['sort'], 1);
    $order_str = ' order by ';
    switch ($column_list[$sort_col-1]) {
        case 'PRODUCT_LIST_MODEL':
            $order_str .= "p.products_model " . ($sort_order ==
'd' ? "desc" : "") . ", pd.products_name";
            break;
        case 'PRODUCT_LIST_NAME':
            $order_str .= "pd.products_name " . ($sort_order ==
'd' ? "desc" : "");
            break;
        case 'PRODUCT_LIST_MANUFACTURER':
            $order_str .= "m.manufacturers_name " . ($sort_order
== 'd' ? "desc" : "") . ", pd.products_name";
            break;
        case 'PRODUCT_LIST_QUANTITY':

```

```

        $order_str .= "p.products_quantity " . ($sort_order
== 'd' ? "desc" : "") . ", pd.products_name";
        break;
    case 'PRODUCT_LIST_IMAGE':
        $order_str .= "pd.products_name";
        break;
    case 'PRODUCT_LIST_WEIGHT':
        $order_str .= "p.products_weight " . ($sort_order ==
'd' ? "desc" : "") . ", pd.products_name";
        break;
    case 'PRODUCT_LIST_PRICE':
        $order_str .= "final_price " . ($sort_order == 'd' ?
"desc" : "") . ", pd.products_name";
        break;
    }
}

$listing_sql = $select_str . $from_str . $where_str .
$order_str;

//HMU ADDED WRAPPER ENDPOINT
// Debug
// print($listing_sql);
$listing_query = tep_db_query($listing_sql);
$row = array();
$rows = array();
$i = 0;
while($row = mysql_fetch_array($listing_query)){
    // MAYBE LOOK AT THIS TRIMMING LATER FOR!!!!!!!!!!!!!!!!!!!!!!
    $rows[$i] = $row[3] . ' ' . trim(trim($row[7], '0'),
'.');
    $i++;
    // Debug
    // printf("Image: %s, Manufacturer: %.0f, Product ID:
%.2f, Name: %.2s, Price: %.2f, Tax: %.2f, Total Price:
%.2f", $row[0], $row[1], $row[2], $row[3], $row[4], $row[5],
$row[7]);
}
mysql_free_result($listing_query);

// returnere pric with tax in Comma Seperated format E.g
"Beloved 39.50,Beuti & The Best 25.99"
$commaSeperatedResult = implode(",",$rows);
return ($commaSeperatedResult);
}
// Use the request to (try to) invoke the service
$HTTP_RAW_POST_DATA = isset($HTTP_RAW_POST_DATA) ?
$HTTP_RAW_POST_DATA : '';
$server->service($HTTP_RAW_POST_DATA);
?>

```


Redevelopment approach service:

productSearch.jws

```

import java.sql.*;
public class productSearch
{
    public String prodSearch(String search)
    {
        // Instantiate variables
        Connection con = null;
        Statement st = null;
        ResultSet rs = null;
        String result = "";

        // Get connection to oscdb using jdbc connector
        try {
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            con =
DriverManager.getConnection("jdbc:mysql://oscdb",
                "root", "");

            // Create and execute statement
            st = con.createStatement();
            rs = st.executeQuery("select distinct
p.products_image," +
                "m.manufacturers_id, p.products_id, pd.products_name,
p.products_price," +
                "p.products_tax_class_id, IF(s.status,
s.specials_new_products_price, NULL) " +
                "as specials_new_products_price," +
                "IF(s.status, s.specials_new_products_price,
p.products_price)" +
                "as final_price from products p left join
manufacturers m using(manufacturers_id)" +
                "left join specials s on p.products_id =
s.products_id, products_description pd," +
                "categories c, products_to_categories p2c" +
                "where p.products_status = '1' and p.products_id =
pd.products_id" +
                "and pd.language_id = '1' and p.products_id =
p2c.products_id" +
                "and p2c.categories_id = c.categories_id" +
                "and ((pd.products_name like '%" + search + "%' or
p.products_model like '%" + search + "%' or
m.manufacturers_name like '%" + search + "%') )" +
                "order by pd.products_name");

            // Get rows
            while(rs.next()) {
                result += rs.getString(4) + " " +

```

```
rs.getString(8).substring(0,rs.getString(8).indexOf(".") +
3) + ",";

    }
    System.out.println(result);

    } catch (Exception e) {
        System.err.println("Exception: " + e.getMessage());
    } finally {
        try {
            if(rs != null)
                rs.close();
            if(st != null)
                st.close();
            if(con != null)
                con.close();
        } catch (SQLException e) {
        }
    }

    // Return result
    return result;

}

}
```


Appendix D - Test client implementation

Hybrid approach test client:

productSearchClient.php

```
<?php
// Pull in the NuSOAP code
require_once('lib/nusoap.php');
// Create the client instance
$client = new
soapclient('http://localhost:85/OSC/joomla/oscommerce-
2.2rc1/catalog/productSearch.php?wsdl', true);
// Check for an error
$error = $client->getError();
if ($error) {
    // Display the error
    echo '<h2>Constructor error</h2><pre>' . $error . '</pre>';
    // At this point, you know the call that follows will fail
}
// Call the SOAP method
$result = $client->call('prodSearch', array('search' =>
'beloved'));
// Check for a fault
if ($client->fault) {
    echo '<h2>Fault</h2><pre>';
    print_r($result);
    echo '</pre>';
} else {
    // Check for errors
    $error = $client->getError();
    if ($error) {
        // Display the error
        echo '<h2>Error</h2><pre>' . $error . '</pre>';
    } else {
        // Display the result
        echo '<h2>Result</h2><pre>';
        print_r($result);
        echo '</pre>';
    }
}
// Display the request and response
```

```
echo '<h2>Request</h2>';
echo '<pre>' . htmlspecialchars($client->request,
ENT_QUOTES) . '</pre>';
echo '<h2>Response</h2>';
echo '<pre>' . htmlspecialchars($client->response,
ENT_QUOTES) . '</pre>';
// Display the debug messages
echo '<h2>Debug</h2>';
echo '<pre>' . htmlspecialchars($client->debug_str,
ENT_QUOTES) . '</pre>';
?>
```

Redevelopment approach test client:

productSearchClient.java

```
package localhost;

public class productSearchClient
{
    public static void main(String[] args) throws Exception
    {

        localhost.ProductSearchServiceLocator locator = new
localhost.ProductSearchServiceLocator();
        ProductSearch productSearch = locator.getProductSearch();

        // make the call
        String ret = productSearch.prodSearch(args[0]);
        System.out.println(ret);
    }
}
```


Appendix E - Service test

Test Case:

Test Case	Input (Type string)	Expected Output (Type string)
TC1	Beloved	Beloved 54.99
TC2	DVD	A Bug's Life 35.99,Beloved 54.99,Blade Runner - Director's Cut 30,Courage Under Fire 29.99,Die Hard With A Vengeance 39.99,Fire Down Below 29.99,Frantic 35,Lethal Weapon 34.99,Red Corner 32,Speed 39.99,Speed 2: Cruise Control 42,The Matrix 30,The Replacement Killers 42,There's Something About Mary 49.99,Under Siege 29.99,Under Siege 2 - Dark Territory 29.99,You've Got Mail 34.99

Table 2: Test case

Hybrid approach service:

TC1 output:

Beloved 54.99

TC2 output:

A Bug's Life 35.99,Beloved 54.99,Blade Runner - Director's Cut 30,Courage Under Fire 29.99,Die Hard With A Vengeance 39.99,Fire Down Below 29.99,Frantic 35,Lethal Weapon 34.99,Red Corner 32,Speed 39.99,Speed 2: Cruise Control 42,The Matrix 30,The Replacement Killers 42,There's Something About Mary 49.99,Under Siege 29.99,Under Siege 2 - Dark Territory 29.99,You've Got Mail 34.99

Redevelopment approach service:

TC1 output:

```

C:\WINDOWS\system32\cmd.exe
C:\jakarta-tomcat-4.0.1\jakarta-tomcat-4.0.1\webapps\axis>java localhost.product
SearchClient Beloved
Beloved 54.99,
C:\jakarta-tomcat-4.0.1\jakarta-tomcat-4.0.1\webapps\axis>
    
```

Figure 13: TC1 output

TC2 output:

```

C:\WINDOWS\system32\cmd.exe
C:\jakarta-tomcat-4.0.1\jakarta-tomcat-4.0.1\webapps\axis>java localhost.product
SearchClient DVD
A Bug's Life 35.99,Beloved 54.99,Blade Runner - Director's Cut 30.00,Courage Und
er Fire 29.99,Die Hard With A Vengeance 39.99,Fire Down Below 29.99,Frantic 35.0
0,Lethal Weapon 34.99,Red Corner 32.00,Speed 39.99,Speed 2: Cruise Control 42.00
,The Matrix 30.00,The Replacement Killers 42.00,There's Something About Mary 49.
99,Under Siege 29.99,Under Siege 2 - Dark Territory 29.99,You've Got Mail 34.99,
C:\jakarta-tomcat-4.0.1\jakarta-tomcat-4.0.1\webapps\axis>
    
```

Figure 14: TC2 output

Test Case results:

Service	Input (Type string)	Expected Output (Type string)	Test Result
Hybrid approach service TC1	Beloved	Beloved 54.99	Passed
Hybrid approach service TC2	DVD	A Bug's Life 35.99,Beloved 54.99,Blade Runner - Director's Cut 30,Courage Under Fire 29.99,Die Hard With A Vengeance 39.99,Fire Down Below 29.99,Frantic 35,Lethal Weapon 34.99,Red Corner 32,Speed 39.99,Speed 2: Cruise Control 42,The Matrix 30,The Replacement Killers 42,There's Something About Mary 49.99,Under Siege 29.99,Under Siege 2 - Dark Territory 29.99,You've Got Mail 34.99	Passed.
Redevelopment approach service TC1	Beloved	Beloved 54.99	Passed.

Appendix E – Service Test

Hybrid approach service TC2	DVD	A Bug's Life 35.99,Beloved 54.99,Blade Runner - Director's Cut 30,Courage Under Fire 29.99,Die Hard With A Vengeance 39.99,Fire Down Below 29.99,Frantic 35,Lethal Weapon 34.99,Red Corner 32,Speed 39.99,Speed 2: Cruise Control 42,The Matrix 30,The Replacement Killers 42,There's Something About Mary 49.99,Under Siege 29.99,Under Siege 2 - Dark Territory 29.99,You've Got Mail 34.99	Passed
--------------------------------	-----	---	--------

Table 3: Test case results