STATUS OF THESIS

Title of thesis: UML-SOA-Sec and Saleem's MDS Services Composition Framework for Secure Business Process Modelling of Services Oriented Applications

I, MUHAMMAD QAISER SALEEM

hereby allow my thesis to be placed at the Information Resource Center (IRC) of Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP
2. The IRC of UTP may make copies of the thesis for academic purposes only.
3. This thesis is classified as

□ Confidential

☑ Non-confidential

If this thesis is confidential, please state the reason:

____

____

The contents of the thesis will remain confidential for _______ years.

Remarks on disclosure:

____

____

	Endorsed by
______________________	______________________
Signature of Author	Signature of Supervisor
Permanent Address: H. No. G-103, Street 3, Phase 4, Officers Colony, Wah Cantt, Distt. Rawalpindi, Punjab, Pakistan.	Name of Supervisor Dr. Jafreezal Bin Jaafar
Date :____________________	Date :____________________

UNIVERSITI TEKNOLOGI PETRONAS

UML-SOA-SEC AND SALEEM'S MDS SERVICES COMPOSITION FRAMEWORK FOR SECURE BUSINESS PROCESS MODELLING OF SERVICES ORIENTED APPLICATIONS

by

MUHAMMAD QAISER SALEEM

The undersigned certify that they have read, and recommend to the Postgraduate Studies Programme for acceptance this thesis for the fullfilment of the requirements for the degree stated.

Signature: ______________________

Main Supervisor: Dr. Jafreezal Bin Jaafar

Signature: ______________________

Co Supervisor: Assoc. Prof. Dr. Mohd Fadzil Bin Hassan

Signature: ______________________

Head of Department: Dr. Jafreezal Bin Jaafar

Date: ______________________

UML-SOA-SEC AND SALEEM'S MDS SERVICES COMPOSITION FRAMEWORK FOR SECURE BUSINESS PROCESS MODELLING OF SERVICES ORIENTED APPLICATIONS

by

MUHAMMAD QAISER SALEEM

A Thesis

Submitted to the Postgraduate Studies Programme

As a requirement for the degree of

DOCTOR OF PHILOSOPHY

INFORMATION TECHNOLOGY

UNIVERSITI TEKNOLOGI PETRONAS

BANDAR SERI ISKANDAR,

PERAK,

MALAYSIA

MARCH 2013

DECLARATION OF THESIS

Title of thesis	UML-SOA-Sec and Saleem's MDS Services Composition Framework for Secure Business Process Modelling of Services Oriented Applications

I, MUHAMMAD QAISER SALEEM

hereby declare that the thesis is based on my original work except for quotations and citations, which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.

Witnessed by

____________________________	____________________________
Signature of Author	Signature of Supervisor
Permanent address:	Name of Supervisor
H. No. G-103, Street No. 3, Phase 4, Officers Colony, Wah Cantt, Dist Rawalpindi, Punjab, Pakistan.	Dr. Jafreezal Bin Jaafar
Date : ____________________	Date : _____________________

DEDICATION

I dedicate this research work to the ***Holy Prophet Muhammad (Peace be upon him),*** to his ***Progeny,*** to his ***Companions,*** to my loving *parents, brother, sisters and especially to my beloved wife, son and daughter*. I also dedicate this research work *to all my friends* for their love, encouragement and great moral support.

ACKNOWLEDGEMENT

I am very thankful to Almighty Allah, the most beneficent, the most merciful and the most gracious, for granting me courage for the successful completion of this work.

First and foremost, my utmost gratitude to my supervisor, Dr. Jafreezal B. Jaafar, whose encouragement, supervision, valuable suggestions and intellectual activities, from preliminary to the concluding level enabled me to complete the whole work. I am extremely grateful to my co-supervisor *Dr. Fadzil B. Hassan* for his keen interest, guidance and moral support.

I am also greatly obliged to Dr. Hafiz Farooq Ahmad and Dr. Mukhtiar Memon for their kind research guidance, discussion and moral support.

In my daily work, I have been blessed with a friendly and cheerful group of fellows. I would like to thank all of them for useful discussions, expedient suggestions and constructive criticism. Specially, I would like to mention Zain Balfagih, Naaman Mussawar, Muzafar Khan, and Fazal-I-Amin.

I would like to appreciate the financial and organizational support from Universiti Teknologi PETRONAS, Malaysia.

I inexpressibly fall short of diction to express my humble obligation to my parents whose hands always rise in pray for my success and because of their whole moral support; I feel my entity at this stage.

Finally, I offer my regards and blessings to all of those who supported me in any respect during my studies and apologize if I have happened to annoy anybody during studies in this great institution.

ABSTRACT

In Service Oriented Architecture (SOA) environment, a software application is a composition of services, which are scattered across enterprises and architectures. Security plays a vital role during the design, development and operation of SOA applications. However, analysis of today's software development approaches reveals that the engineering of security into the system design is often neglected. Security is incorporated in an ad-hoc manner or integrated during the applications development phase or administration phase or out sourced. SOA security is cross-domain and all of the required information is not available at downstream phases. The post-hoc, low-level integration of security has a negative impact on the resulting SOA applications. General purpose modeling languages like Unified Modeling Language (UML) are used for designing the software system; however, these languages lack the knowledge of the specific domain and "security" is one of the essential domains. A Domain Specific Language (DSL), named the "UML-SOA-Sec" is proposed to facilitate the modeling of security objectives along the business process modeling of SOA applications. Furthermore, Saleem's MDS (Model Driven Security) services composition framework is proposed for the development of a secure web service composition. Being able to express security objectives in a widely used design notation like UML, helps to save time and effort during the implementation and verification of security in SOA applications. As a proof of concept, the research work is projected on a case study of the real world SOA application from the healthcare domain.

ABSTRAK

Dalam arkitektur berdasarkan servis (*Service Oriented Architecture, SOA*), perisian aplikasinya ialah gabungan servis-servis yang berkaitan perusahaan dan arkitektur. Aspek keselamatan merupakan yang terpenting ketika proses reka bentuk, pembangunan dan operasi SOA tersebut. Walaubagaimanapun, analisa semasa terhadap pendekatan pembangunan perisian telah mendedahkan bahawa aspek keselamatan pada reka bentuk sistem seringkali diabaikan. Aspek keselamatan ini digabungkan secara *ad-hoc*, secara integrasi semasa fasa pembangunan perisiandan fasa pentadbiran ataupun dari sumber luar. Keselamatan SOA ialah domain yang bersilang dan setiap maklumat yang diperlukan tiada dalam fasa hiliran. Aspek keselamatan yang dibuat secara *post-hoc* dan integrasi pada tahapan rendah mempunyai kesan buruk dalam menghasilkan perisian SOA. Bahasa pembentukan yang mempunyai tujuan umum seperti *Unified Modeling Language (UML)* untuk mereka bentuk sistem perisian, walaubagaimanapun, bahasa-bahasa ini mempunyai pengetahuan yang terhad tentang domain yang spesifik dan aspek keselamatan adalah salah satu domain yang utama. *Domain Specific Language (DSL)*, yang diberi nama "UML-SOA-Sec" dicadangkan untuk membantu membentuk objektif keselamatan sepanjang proses perniagaan membentuk perisian SOA. Tambahan pula, servis gabungan rangka kerja Saleem's MDS (*Model Driven Security*) juga dicadangkan untuk membangunkan gabungan servis web yang selamat. Ia mampu untuk memberi penekanan pada objektif keselamatan dalam kegunaan notasi reka bentuk secara meluas seperti UML, ia juga membantu untuk menjimatkan masa dan tenaga sewaktu proses perlaksanaan dan pengesahan aspek keselamatan dalam perisian SOA. Sebagai bukti kepada konsep ini, kajian ini juga dilaksanakan dalam salah sebuah perisian SOA iaitu dari domain kesihatan.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF GRAPHS

LIST OF ABBREVIATION

SOA	Service Oriented Architecture
WS	Web Service
SOSE	Service-Oriented System Engineering
DSL	Domain Specific Language
UML	Unified Modelling Language
BPMN	Business Process Model and Notation
BPEL	Business Process Execution Language
XPDL	XML Process Definition Language
JPDL	jBPM Process Definition Language
MDSD	Model Driven Software Development
MDE	Model Driven Engineering
MDS	Model Driven Security
MDA	Model Driven Architecture
MOF	Meta Object Facility
PIM	Platform Independent Model
PSM	Platform Specific Model
ISM	Platform Independent Model
XML	Extensible Markup Language
XMI	XML Metadata Interchange
SOAP	Simple Object Access Protocol
WSDL	Web Services Description Language
UDDI	Universal Description Discovery and Integration
HTTP	Hypertext Transfer Protocol
SLA	Service Level Agreement
COTS	Commercial Off The Shelf
HTTP	Hypertext Transfer Protocol
OMG	Object Management Group
W3C	World Wide Web Consorium

CHAPTER 1
INTRODUCTION

1.1 Introduction

Today's Information Technology (IT) environment is network/Internet centric such as Service Oriented Architecture (SOA), Cloud, and Software as a Service (SaaS) which offer the IT agility demanded by business [1, 2]. In an SOA environment, software applications are deployed over the Internet as a service. To support business ventures, these services are integrated/composed within and across organizations to form Internet-based systems and perform cross application transactions [3]. The paradigm of SOA promises inter-operability and integration ensuring the availability of resources in the form of services over the network. However, it is full of daily virus alerts, malicious crackers and the threats of cyber terrorism [1, 2]. When attacks on the system increase, it is probable that an intrusion can be successful [4]. The security violation causes losses; hence, it is necessary to secure the whole SOA system.

Although there is a great acceptance of web services technology in the market for SOA applications; however it lacks the modelling technique that can guarantee the Quality of Service (QoS) for the development of the SOA application. There is no sufficient support for modelling security objectives in SOA [5, 6]. During business process modelling, which is performed by a business domain expert, concentration is towards modelling the business process in a way that functional correctness is modelled; however, usually the notion of security is often neglected. It may happen due to many reasons e.g. a business domain expert is not a security expert [4, 7], and no currently available business process modelling language has the ability to capture security goals [8]. Furthermore, a security model and system model are disjointed and expressed in different ways i.e. security model is represented in a structured text while system model is represented in a graphical way in a modelling language like Unified Modelling Language (UML) [4].

A Domain Specific Language (DSL) named “UML-SOA-Sec” is proposed for the modelling of security objectives along business process modelling of SOA systems. The proposed DSL is based on the essential security objectives for an SOA environment. Metamodel of the proposed DSL is defined illustrating the security objectives and the security mechanisms through which these security objectives would be realized. Afterwards, a UML profiling mechanism is used for the definition of these security objectives as stereotypes in a modelling language. After the definition of the domain specific UML-profile, a general-purpose modelling tool can easily be specialized and these domain specific stereotypes are made available at the modelling level in the form of annotation.

In an SOA environment, software applications are basically a composition of services; Saleem’s MDS services composition framework is proposed for the development of a secure web service composition. The aim is the identification and organization of the basic steps/phases for the service composition and identification of those steps where security would be modelled along the business process modelling. During this research work, the UML-SOA-Sec is used for the modelling of the security objectives along the business process modelling.

As a proof of concept, a security annotated business process model is created and a prototype is developed based on the real world example of an SOA environment related to healthcare domain. Comparative analysis and evaluation of the proposed work with the related work is performed.

1.2 Background

Businesses today are challenged to manage their increasingly complex, interconnected Information Technology (IT) landscapes. Modern network / Internet centric IT architectures and paradigms, such as SOA, Cloud and SaaS (Software as a Service) offer the IT agility demanded by businesses [1]. SOA is currently the best available solution for enterprises for achieving interoperability, agility and other goals. Success stories from industry (e.g. Amazon and Google) show high acceptance potential in using the SOA for distributed system development and it appears to be the most promising paradigm for distributed computing application design, development and deployment [9]. The SOA align businesses and IT, and it provides many business as well as technical benefits [1]. The SOA paradigm promises *"1) rapid application development to significantly enhance corporate agility 2) automated business processes and 3) multi-channel access to applications"* [10]. The SOA paradigm utilizes services as a fundamental element for developing applications [11] and makes the application development easy by coupling services over the intranet or via the Internet [12] and it has changed the Internet from being just a repository of data to a repository of services [13]. SOA is a design model with a concept of encapsulating application logic within services that interact via a common communication protocol [14, 15]. Furthermore it is an architectural style in which software applications are comprised of loosely coupled reusable services by integrating them through their standard interface. Services are independent of language, platform and location and may be locally developed or requested from the provider. A business process can be realized as a runtime orchestration of a set of services. Software applications are often comprised of numerous distributed components such as databases, web servers, computing nodes, storage nodes etc. and these components are distributed across different independent administrative domains. Services are used but not owned by the user and they reside on the provider side [16-18]. SOA is also called a "*Find, bind and invoke paradigm*" [3, 19].

Currently, most for the enterprises develop their Web Information Systems (WISs) using web service technology by composing web services which may be geographically located at different sites using the SOA paradigm [20]. The area of web services composition has gained an interest in the web service community; however, most of the research work addresses implementation and execution issues. Therefore, many composition languages have been proposed in recent years such as Business Process Execution Language (BPEL), XLANG (X Language), Web Services Flow Language (WSFL) and Web Service Choreography Interface (WSCI) etc. to name few of them. However, these languages are not related to the early stages of the system development [21]. Furthermore, few methods/frameworks [21-24] are presented for services composition; however, they do not deal with security.

Many software engineering approaches are used for the development of SOA systems; among them, the Model Driven Software Development (MDSD) is one of the most promising approaches. The Object Management Group (OMG) has presented a framework known as the Model Driven Architecture (MDA) [25], which is considered as an implementation of Model Driven Engineering (MDE). In the MDA framework, software systems are modelled using a general purpose modelling language like UML, as a Platform Independent Model (PIM), and then it is transformed into other PIM or Platform Specific Model (PSM). In the MDA framework, rather than just a visual aid, models are considered as essential parts of the software definition [10, 26]. The MDS is a specialization of the MDSD to the domain of security. The crucial part of this specialization is concerned with the modelling language [27]. Many researchers have proposed many security DSLs by extending existing modelling languages like the UML, focusing different aspects of software development. Automatically developing software applications enriched with security configuration is a topic of interest among the research community and many research groups are trying to address the security problems for software applications [12, 28-33].

For security enhancement in a business process model, two types of descriptions are used, formal and informal. An informal description is a text-based description written in a natural language; whereas a formal description uses a formal language with a defined syntax. The informal specification may be problematic because it is written in a natural language; hence, the statements may be ambiguous and different readers may interpret them differently [34]. That's why this work chooses the formal definition of security enhancement, which is easy to define and validate and is known as a high level security requirement.

For an enterprise, business processes are the key to maintain competitiveness, since, business processes are the ability of an enterprise to describe, standardize and adapt the way it reacts to business events. They also illustrate how an enterprise interacts with partners, suppliers, customers, competitors etc. [35]. Business process modelling is the most appropriate layer to describe security requirements and to evaluate risks [20]. The actual stakeholder of the business process is the business domain expert. He must specify the security objectives during the business process modelling at a very abstract level. If it is left to the security specialist, who is a technical person, he will specify them at the technical level resulting in the loss of domain knowledge about the compliance regulations or its refinement, which is not intuitively traceable [30]. Empirical studies show that a business domain expert is able to specify security objectives at high levels of abstraction i.e. during designing of the system [4]. Usually, the business domain expert is involved in modelling the functional diagram and application architecture. It is a fact that he/she is not a security specialist. He/she is familiar with the common security concept; however, he/she does not have the expertise of how these security concepts are realized. He/she can define a business process model as a UML activity diagram and add security objectives as a UML stereotype [30, 36]; however, he/she does not concentrate on security mechanisms and security technologies. Those are later on accomplished by the architectural team. Many researchers like D. Basin [37], M. Hafner [38] F. Satoh [39] etc. deal with similar processes to realize the security modelling concepts.

1.3 Motivation

UML models are constructed during software development; however, after some time, they become abandoned. With the passage of time, software has to evolve; normally developers change the code instead of model. As a consequence, models become invalid and useless because they do not represent the actual software system. Since these models are constructed by using general purpose modelling languages like the UML, only a small part of the code is generated from the model and additional parts of the implementation are added manually into the code generation. As a result, general purpose modelling languages like the UML do not raise the productivity to a sufficient level because they lack the domain-specific concepts [40].

During recent years, an important goal of the software researchers has been to develop techniques where domain concepts are modelled in terms of design intent rather than the underlying implementation environment [10]. Extending a general purpose modelling language according to a particular domain and defining the Domain Specific Language (DSL) is a common practice e.g. creating UML extensions according to specific business domains like data warehousing [41], business intelligence [42] and real-time systems [43], system aspects like security [32, 44] or concrete technologies like the Application Programme Interface (API) for different programming languages [45] etc. The DSL is one the key concepts in Model Driven Engineering (MDE) [10].

As compared to the general purpose modelling languages, DSLs offer substantial gain in ease of use and expressiveness according to the specific domain for which they are developed. DSLs result in several benefits such as considerable gain in productivity, reduction in maintenance cost and reducing the required domain specific expertise. DSLs are also called application oriented, special purpose, specialized or task specific languages. Appropriate notions related to the specific domain are usually beyond the notation offered by general purpose modelling languages. General purpose modelling languages do not render the superfluous of DSLs and they are very clumsy for tasks that can benefit from the integration of the domain-specific restrictions [46]. DSL development requires language development expertise as well as domain knowledge [45].

The definition of a security DSL for SOA systems and its incorporation in a business process model is a challenging task due to many reasons [47]:

- There is not a clear identification of security objectives to be modelled for SOA applications.
- There is absence of notations to express the security objectives.
- There is difficulty in integrating security objectives into a business processes model.

UML and Business Process Model and Notation (BPMN) are the industry standards for business process modeling [48]. Being able to express security objectives in a widely used design notation like the UML or BPMN [48] for SOA systems, helps to save time and effort during the implementation and verification of security in the system [49]. Furthermore, specifying security objectives at an abstract level helps the architectural team in choosing different and potentially better, security mechanisms e.g. biometric devices such as retina scanners, fingerprint readers etc. to meet the real underlying security requirements [2].

SOA applications are basically composition of services which are scattered across the Internet. Several web services composition frameworks/methods are proposed [23, 24, 50, 51] containing several different combination of steps/phases (ranging from four to thirteen). However, there are no clear identifications of the most necessary steps/phases for service composition framework. Furthermore notion of security is neglected in almost all of these frameworks i.e. security is not defined during the business process modelling of SOA applications developed through these frameworks.

1.4 Research Problems

A thorough literature study reveals the following research problems:

1.4.1 Problems in Current SOA Security Practices

Security must be unified with the software engineering process; however, in practice, it is considered an afterthought and implemented in an ad-hoc manner i.e. during the implementation phase or during the system administration phase or it is outsourced [4]. In many cases it is left to the developer and added when the functional requirements are met or at the time of the integration of distributed applications. The development of SOA system is very complex; which a developer alone cannot meet anymore; moreover, security is difficult to manage and costly to maintain [12, 52]. Moreover, in practice, security requirements are implemented in the system with a programming language dependent handcrafted fixed code; such an inflexible code cannot meet the unforeseen challenges of an SOA environment e.g. changes in business logics, workflow variations and patchy platform technologies [29]. Furthermore, SOA applications are cross-domain and coupled over various network technologies and protocols; just adding a security code to software applications is not a realistic approach because all the required security information is not available at the downstream phases[12, 53]. This approach degrade implementing and maintaining security of the system [37].

Furthermore security requirements are specified in a "*non-formalized*" way by the business department normally as an unstructured text. If these security specification are not understood by the IT security department, a complicated and error prone coordination process between both departments arises, this result in a loss of requirement sovereignty by the business department which is the owner of the application [54].

Business process modelling is the most appropriate layer to describe security requirements and to evaluate risks [20]. Empirical studies shows that those, who model the business process i.e. business domain experts are able to specify security requirements at high levels of abstraction i.e. while designing the system [4]. During business process modelling, which is performed by a business process expert, concentration is towards modelling the business process in a way that functional correctness is modelled and the notion of security is often neglected. It may happen due to many reasons e.g. the business domain expert is not a security expert [4, 7], and no currently available business process modelling language has the ability to capture security goals [8]. Furthermore system models and security models are disjointed and expressed in different ways i.e. system a model is represented in a graphical way in a modelling language like a UML while a security model is represented as a structured text [4].

During the past few years, several SOA security protocols, access control models and security implementations have emerged to enforce the security goals [12, 30]; however, the focus of these security standards and protocols are towards the technological level; which does not provide a high level of abstraction and mastering them is also a daunting task [20, 55]. This approach will leads to security vulnerabilities, which justify increasing effort in defining security in pre-development phases, where finding and removing a bug is cheaper [56].

1.4.2 Lack of Expression for Domain Knowledge in General Purpose Modelling Languages

Currently business process analysts express the business logic of the SOA applications with the help of a general purpose modelling language like a UML or BPMN [57]. Currently available business process modelling languages do not have the ability to capture security goals [8]. A general purpose modelling language has a broader scope and there may be a situation where it is not appropriate for modelling of some specific domains e.g. security, real-time etc. Furthermore, there may be situations when the syntax and semantics of the general purpose modelling language's elements are not able to express the specific concepts of particular systems or there may be a situation when these element may be customized or restricted which is normally too general and too abundant [58].

For modelling a specific domain, general purpose modelling languages have three main limitations: lack of semantics, lack of visualization and lack of abstraction while preparing a business process model [59]. The same is the case for the security i.e. these general purpose modelling languages do not support the specifications of security requirements [8, 33].

1.4.3 Unclear Security Objectives for an SOA Environment.

There are numerous SOA security objectives that may differ for each stakeholder like vendors, security experts, consultants, business process experts etc. Similarly, security objectives can be related to some specific business case, technology, governance, deployment etc. Unclear security objectives result in unclear security implication which is cited as one of the most important issues that limit the SOA benefits and hence slow down its adoption[1]. Furthermore, current model-driven approaches where business processes are enhanced with security objectives, do not describe the consistent selection of appropriate security objectives [20].

1.4.4 Security is not Defined During the Business Process Modelling in Services Composition Frameworks

SOA applications are basically a composition of services which are scattered across the Internet. Several web services composition frameworks/methods are proposed [23, 24, 50, 51]; however, the notion of security is neglected in almost all of them i.e. security is not defined for services composition modeling. These frameworks just describe the different combinations of steps/phases of services composition and do not focus the definition of security objectives during the business process modeling of SOA applications developed through these frameworks.

1.5 Aims and Objectives

The aim of the dissertation is to specialize the concept of the MDE to Model Driven Security Engineering (MDSE) for SOA applications

The following are the objectives of this research work:

1 Modeling of security objectives along the business process modelling of SOA applications.

2 Development of MDS services composition framework where security objectives are modelled during the business process modeling of services composition.

3 Comparison and evaluation of security enhanced business process model using the proposed approach with the related approaches.

1.6 Research Questions

Keeping in view the research problems, this research work has identified the following research questions:

1 What are the essential security objectives to be modeled during the Business Process modeling for SOA applications?

2 How can the general purpose modelling language be enriched to specify security objectives in a Business Process Model of an SOA application in a formalized manner?

3 What are the essential steps/phases for a services composition framework? And at which steps/phases of the services composition framework, security objectives would be defined/modelled?

1.7 Proposed Solutions

To address the research questions, the following solutions are proposed:

- A Domain Specific Language named "UML-SOA-Sec" has been proposed comprised of the most essential security objectives to be modelled during the business process modeling of SOA applications. General purpose modelling language UML has been extended and security stereotypes are defined for each security objective. MagicDraw UML modeling tool is used which support the definition and usage of stereotype. UML-SOA-Sec facilitates the business process experts in modelling security objectives along the business process modelling of SOA application.

- Saleem's MDS services composition framework has been proposed for the development of secure web services composition. Proposed framework is comprised of the four most essential steps for web services composition. In the proposed framework, security objectives are defined along the business process modelling, which is performed using UML activity diagram. UML-SOA-Sec is used for the definition of security objectives.

A domain expert is facilitated in defining the security objectives and this security annotated business process model is used as an origin of a Model-Driven software development approach to generate concrete security for SOA applications [54].

1.8 Thesis Organization

After describing the introduction, background, motivation, problem statements, aims and objectives, and methodology, the remainder of the thesis is organized as follows:

Chapter 2 introduces the background knowledge related to our research work. First it introduces the SOA environment and the Web Services. Thereafter, the chapter illustrates the services composition and different standards/languages used for the web services composition. Then it illustrates the different concepts like the business process modelling, MDSD, DSL and UML-Profile. Then the chapter highlights the security for the SOA systems. After that, the chapter illustrates the literature consulted in order to find out the essential security objectives for the SOA applications. The chapter later on discusses the related work regarding the two areas of focus of this dissertation i.e. security modelling during early phases of software development and web services composition frameworks. End of the chapter described the tools and technologies used for the prototype implementation of the case study.

Chapter 3 discusses research methodology of the whole research work. It starts with illustrating the different research methods used in the software engineering research. After that, it described in detailed the present work research activities. Afterwards it describes in detailed the research methods used during this research work for the validation of the proposed work. At the end, there is detailed discussion about the research data analysis.

Chapter 4 discusses the proposed work in detail. First, it illustrates an overall view of the whole process of the DSL definition. Afterwards, it describes the abstract syntax (metamodel) and concrete syntax of the proposed DSL, “UML-SOA-Sec”. Finally the chapter describes the “Saleem’s MDS services composition framework” and illustrates in detail the different steps/phases involved in it.

Chapter 5 describes the case study related to the healthcare scenario, used for the validation of the proposed work. The chapter starts with the introduction of the healthcare scenario and suitability of the SOA environment for healthcare systems. Thereafter, the chapter discusses the importance of security for healthcare systems and describes the security objectives of healthcare systems. Afterwards, the chapter describes in detail the case-study and has a thorough discussion about the different security objectives for the healthcare scenario under this study. Thereafter, the business process model is presented through the UML Activity diagram where security is incorporated using the proposed DSL “UML-SOA-Sec”.

Chapter 6 presents the whole process of secure composite application development, deployment and its usage. It starts with the description of UML deployment diagram. Afterwards it illustrates the EIS BPEL workflow diagram, JBI service assembly unit and composite application deployment. At the end there is a detailed description about the scenario when a client application wants to access the composite application and how security checks are ensured.

Chapter 7 presents the analysis and results of the dissertation. In this chapter comparative analysis of the proposed work is perform to obtain the qualitative results. At the start, it presents the comparative study of the proposed DSL UML-SOA-Sec with the related work. Afterwards, it described the comparison of proposed Saleem’s services composition framework with the other frameworks for services composition.

Chapter 8 presents the evaluation and discussion of the dissertation. Initially evaluation process is presented in which a quantitative method (survey) is used to evaluate the “UML-SOA-Sec” to obtain the quantitative results. Afterwards, it presents the discussions about, how the research questions of the dissertation are addressed. At the end of the chapter, discussions are presented after combining both kinds of results i.e. quantitative and qualitative regarding the significance of the approach used in the proposed DSL “UML-SOA-Sec” over the approaches used in other DSLs to annotate the security in a business process diagram.

Chapter 9 concludes the work by summarizing the main contributions and findings of the research work, the limitations of the research work and some possibilities for future research and development.

A list of publications during this research work is provided at the end of the thesis.

Appendix ‘A’ provides the questionnaire used to conduct the survey.

Appendix ‘B’ provides the respondent’s feedback, organized in tables according to model of a particular researcher.

Appendix ‘C’ provides the list of respondents of the survey.

Appendix ‘D’ provides the sample code of the prototype.

CHAPTER 2
LITERATURE REVIEW

2.0 Chapter Overview

This chapter presents the background knowledge, literature review and work related to this research work. A comprehensive discussion is provided about the basic concepts involved in this work i.e. SOA, Web services, Web services composition and its standards/languages, SOA Security, Business Process Modelling, MDSD, DSL and language extension mechanisms. Afterwards there is a detailed discussion about the literature consulted in order to find out the essential security objectives to be modelled for the SOA applications. It starts with describing the term security objectives, and then the chapter describes the security objectives present in the related work. Afterwards, the chapter divides the literature into two groups (1) security objectives-general and (2) security objectives-SOA. A detail discussion is provided regarding finding the essential security objectives to be modeled for SOA applications. At the end, the related work is presented in the area of DSL definition and the services composition framework followed by the tools and technologies used during the research work.

2.1 Service Oriented Architecture (SOA)

Businesses today are challenged to manage their increasingly complex, interconnected Information Technology (IT) landscapes. Modern network / Internet centric IT architectures and paradigms, such as SOA, Cloud and SaaS (Software as a Service) offer the IT agility demanded by businesses [1]. SOA is currently the best available solution for enterprises for achieving interoperability, agility and other goals. Success stories from industry (e.g. Amazon and Google) show high acceptance potential in using the SOA for distributed system development and it appears to be the most promising paradigm for distributed computing application design, development and deployment [9]. The SOA align businesses and IT, and it provides many business as well as technical benefits [1]. Business benefits include agility, cost savings, reusability, business-led-ownership etc. whereas, technical benefits include reduced complexity, easier changes easier reuse, easier integration of new systems and provision of a consolidated view etc. The Service-oriented approach is complementary to other development approaches that preceded it e.g. Object Oriented, Component Based Development (CBD) etc. [60]. Large computing vendors like Microsoft [11], IBM [61], HP [62, 63], SAP [64, 65] , Oracle [66-68], Cisco [69] and many others are aggressively marketing hardware, software, tools, languages, frameworks, standards and services that support SOA implementation for various application areas like healthcare, defense, banks, manufacturing companies, finance, trading etc. [60]. Customers are embracing SOA as a way to successfully reach a previously unachievable level of interoperability among systems and agility in business practices [16]. The SOA paradigm promises *"1) rapid application development to significantly enhance corporate agility 2) automated business processes and 3) multi-channel access to applications"* [10].

For the development of SOA systems; Service-Oriented System Engineering (SOSE) is used that comprises a set of software engineering techniques needed for the analysis, modelling, specifications, creation, testing, debugging, monitoring, and governance of SOA applications [70, 71]. The SOA paradigm utilizes services as a fundamental element for developing applications [11] and makes the application development easy by coupling services over the intranet or via the Internet [12] and it has changed the Internet from being just a repository of data to a repository of services [13]. SOA is a design model with a concept of encapsulating application logic within services that interact via a common communication protocol [14, 15]. Furthermore it is an architectural style in which software applications are comprised of loosely coupled reusable services by integrating them through their standard interface. Services are independent of platform, location and language; moreover they may be local or requested remotely. A software application based on a business process can be realized as a runtime orchestration of a set of services. Software applications are often comprised of numerous distributed components such as databases, web servers, computing nodes, storage nodes etc. Services are used but not owned by the user and they reside on the provider's side [16-18]. The SOA is also called a "*Find, bind and invoke paradigm*" [3, 19] as shown in Figure 2.1

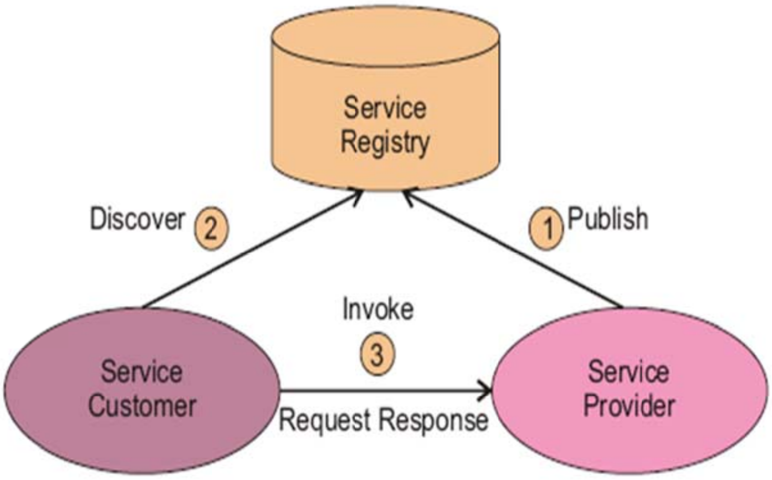


Figure 2.1: Collaboration of Services in an SOA environment [3, 19]

The basic building block of an SOA paradigm is a service. "*A service is an implementation of a well-defined piece of business functionality, with a published interface that is discoverable and can be used by service consumers when building different applications and business processes*" [72]. These services are regarded as autonomous, platform-independent, computational elements that can be described, published, discovered, orchestrated, and programmed using standard protocols for building interoperating applications [70].

Technically, a service consists of three parts: 1) Contract: It provides formal and informal specification of the service. 2) Interface: It is a technical representation of the operations provided by the service which a client can invoke. 3) Implementation: This is the actual logic of service [19] as shown in Figure 2.2.

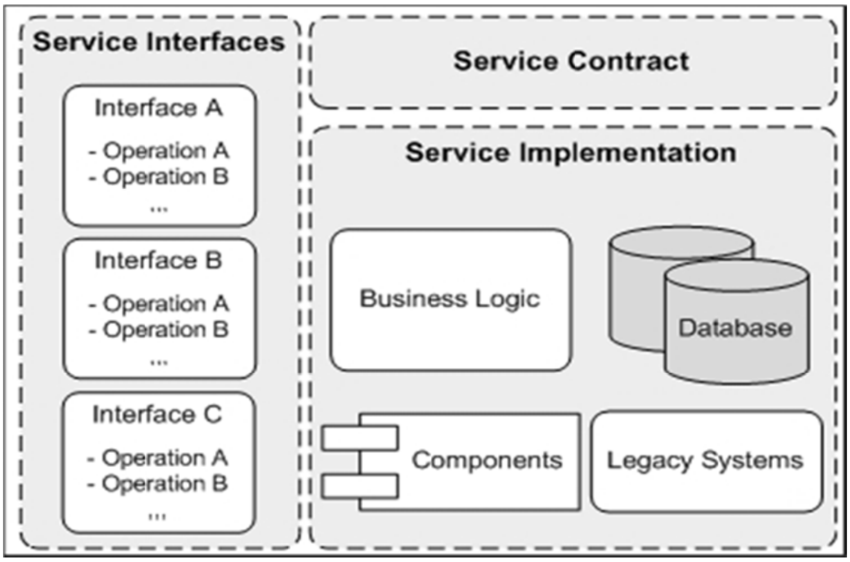


Figure 2.2: Essentail Service Elements [19]

An SOA paradigm can be implemented with different technologies like the Common Object Request Broker Architecture (CORBA), Web services and JINI (pronounced GEE-nee; loosely derived from the Arabic for magician) etc. as represented in Figure 2.3. However, Web services technology is a widespread accepted instantiation of an SOA [18, 73] and due to its broad industry acceptance, web services and SOA names are used inter-changeably [10] which is a common misunderstanding.

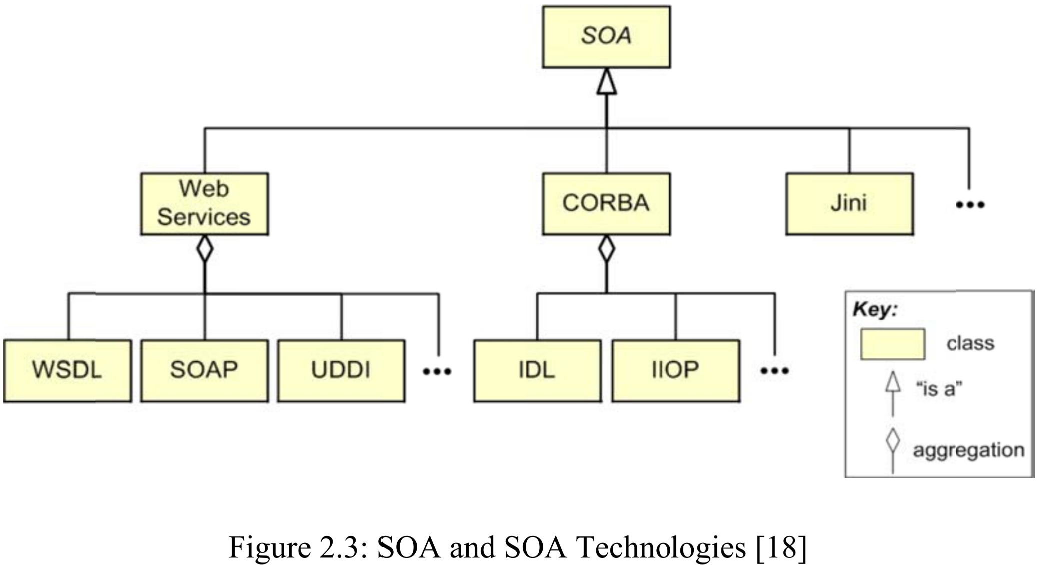


Figure 2.3: SOA and SOA Technologies [18]

Post development stages express the real strength of the SOA paradigm, when a new business process can be realised by just composing existing services and a new application is developed by just assembling existing reusable services [10].

2.2 Web Services (WS)

Web services are defined as "*self-contained, modular units of application logic which provide business functionality to other applications via an Internet connection*" [73]. Web services are also defined as : " *network (Internet) based modular applications designed to implement an SOA, and support interoperable, loosely coupled, integration of heterogeneous applications*" [13]. In Web services technology, software applications are developed by integrating different Web services either newly built or legacy applications; this is accomplished both within and across organizational boundaries by avoiding difficulties due to heterogeneous platforms and programming languages by exploiting the Extensible Markup Language (XML) and the Internet technologies [57, 73]. An overall business solution comprises many reusable services, where each service is designed, developed and deployed independent of the others; moreover, a business process is realized via runtime orchestration of a set of loosely coupled services [17]. Web services enable dynamic connections and automation of business processes within and across enterprises for EAI (Enterprise Application Integration) and B2B (Business-to-Business) integration [73]. The basic idea behind Web services technology is to exploit the Internet and XML technologies and to develop applications by integrating Web services which are published, located and invoked over the web.

Web services are built on standards which are supported by major software vendors [73]. Basic standards for the Web services technology are the Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL) and Universal Description Discovery and Integration (UDDI) which are XML based and typically conveyed using the Hypertext Transfer Protocol (HTTP) [13, 19, 74, 75].

SOAP is used for information exchanges in a distributed and decentralized environment and it is implemented with the help of typed message exchange and remote invocation [57]. SOAP defines the interface of the Web service which is used to invoke it by other service [73]. The WSDL is used for the Web service description in an XML format on a standard message layer e.g. SOAP. In a WSDL document, Web services are defined as a network port or endpoint. A port is associated with a network address and a service is composed of ports that provide operations which are of four types (1) sends a message (one-way), (2) receives and sends a message (request-response), (3) sends and receives a message (solicit-response) and (4) receives a message (notification) [57].

In the WSDL, abstract definitions of ports and messages are separated from concrete network deployment, which allows the reuse of messages and ports. Basically messages are abstract descriptions of data being exchanged and ports are the abstract collections of operations [57]. The WSDL provides a function-centric description of Web services covering inputs, outputs and exception handling [73]. The UDDI is a set of services which supports the description and discovery of three things: 1:) Web service, 2) Web service providers, and 3) technical interfaces used to access the Web services. UDDI is used to build the yellow pages for the Web service [57].

Services provided through the UDDI can be retrieved either programmatically using APIs (Application Programmer Interface) or manually using URLs (Universal Resource Locator) that is provided by the UDDI registries [76]. Figure 2.4 shows the layers of Web services standards.

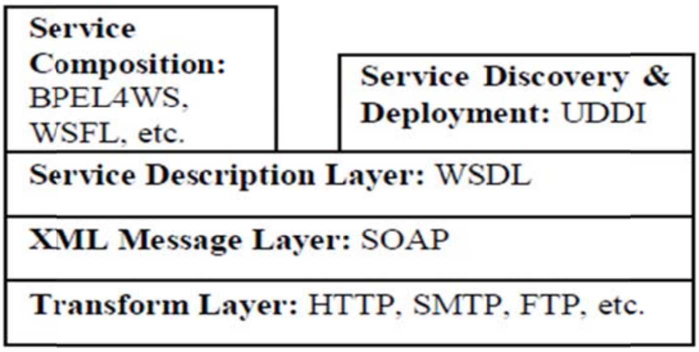

Service Composition: BPEL4WS, WSFL, etc.
Service Discovery & Deployment: UDDI
Service Description Layer: WSDL
XML Message Layer: SOAP
Transform Layer: HTTP, SMTP, FTP, etc.

Figure 2.4: Layers of Web Services Standards [74]

In its most basic form, the working is as follow: a provider publishes a description of its service in the WSDL. The service requestor accesses the descriptions using the UDDI registry and requests the execution of the service by sending a SOAP message to it.

2.2.1 Web Services Specification Stack

Services are distributed over the network; their description finding and access is a big issue. Organizations like OASIS, W3C, OMG etc. and companies like SUN, IBM, Microsoft etc. have presented a standardized way to describe, locate and access services that are distributed over the network known as the Web Service Specification Stack as shown in Figure 2.5.

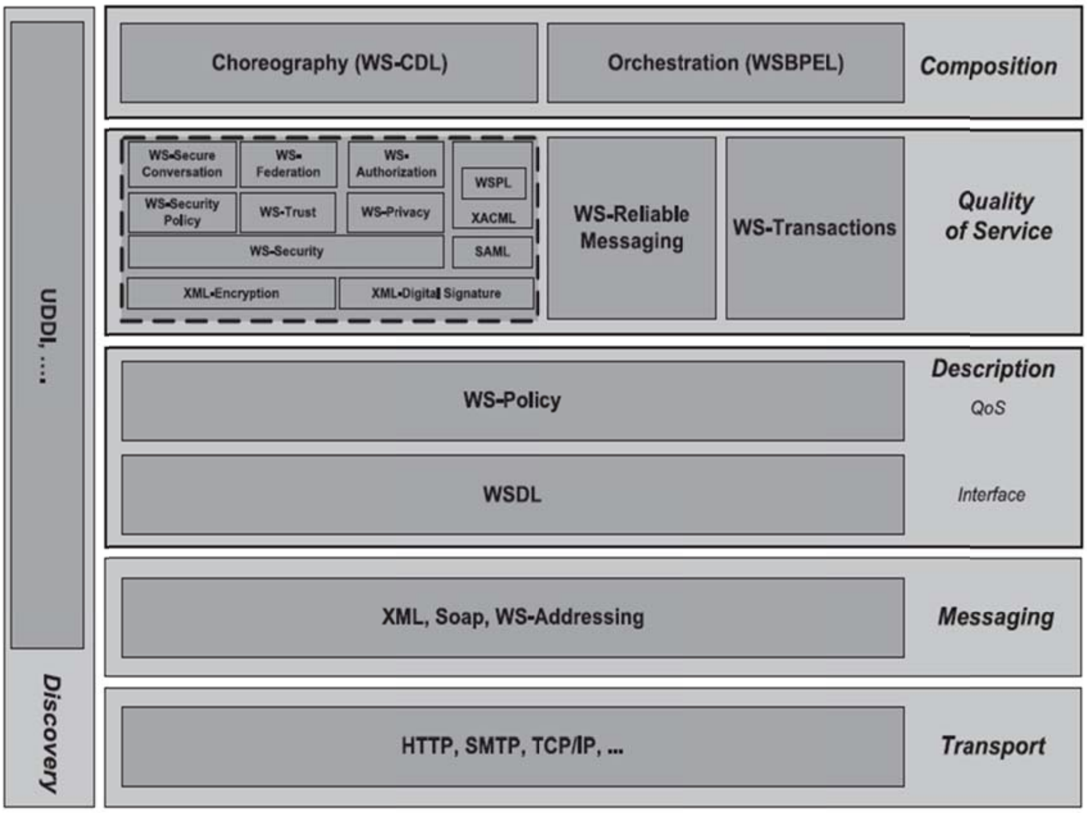


Figure 2.5 : Web Service Specification Stack [38]

Since a variety of different protocols are used in SOA environments, this stack provides a bird's-eye view of them and categorizes them according to their functionality and level of abstraction. Below is a brief description about the six layers described in the stack and standards used in the particular layer; to get the basic understanding about the working of a Web services based SOA environment [38].

1. Transport layer: Web services based applications can be accessed using any common communication protocol e.g. TCP/IP, HTTP etc. which means that a Web services platform works on an application layer and is known as transport neutral messaging architecture.

2. Message layer: In this layer, three standards are used namely the XML, SOAP and WS Addressing. The XML is a common platform independent data format that can be used to represent structured data types independent of any programming language

used to parse it. SOAP messages are used for communication among the Web services. SOAP messages contain the addressing information by incorporating HTTP-specific data in the SOAP message-header to make it transport neutral. WS-Addressing facilitates end-to-end messaging by providing a standard way of representing the Web services end point in messages.

3. Description Layer: This layer describes the interface of a service using the WSDL. The WSDL includes the metadata to construct the message body. Other metadata specification includes the XML schema, WS-policy and WS–MetadataExchange.

4. Quality of service layer: This layer describes the non-functional aspect of the Web-services e.g. security, reliability of message delivery and transactional support, metadata acquisition and orchestration etc. The WS-Transaction is used for the transaction management activities and WS-Reliable Messaging is used for reliable message delivery among the Web services. There are numerous Web services security standards as can be seen in the diagram.

5. Services Composition: Web services are scattered across the Internet; rather than each service invoking each other by using message exchange patterns like SOAP, a mechanism is developed to compose more complex interactions among Web services. At the composition layer, the execution order of a service invocation and its interaction pattern is defined. Services are composed using services composition languages/standards. The composition consists of the invocation of Web services in the form of Choreography or Orchestration. In Orchestration, there is a central control which describes the execution order of the Web services; while, in Choreography, there is not a central control and Web services interact with each other in a peer-to-peer fashion. The WS-BPEL (Web Services Business Process Execution Language) is used for orchestration while the WS-CDL (Web Services Choreography Description Language) is used for the choreography.

6. Service Discovery: In SOA environments, services are scattered over the Internet and their discovery is very important in a distributed environment. Potential requestors must be able to search/discover service descriptions according to their business needs. This is achieved by repositories which provide records of the services called registries. Services are registered and discovered through a registry and a UDDI is used for the specification of Web services discovery.

2.2.2 Web Services Security Standards

Different techniques, algorithms and mechanisms are used to counter a specific threat in SOA based information systems. Web services security standards leverage them and abstract from the specific implementation details. In Web services based SOA environment, popular security standards are the Web Services Security (WS-Security), XML-Digital Signature, XML-Encryption, Security Assertions Markup Language (SAML) and eXtensible Access Control Markup Language (XACML) [52]. The following is a brief description of these standards.

WS-Security is an OASIS standard which was originally developed by IBM, Microsoft and VeriSign. WS-Security provides the comprehensive policy for message protection and also provides a few further standards for different tasks as cleared from the names of standards like the WS-Authorization, WS-Privacy, WS-Trust, WS-Federation, WS-Policy, and WS-Secure Conversation. Two more standards are used in the Web services community are the SAML and the XACML. The SAML standard is an XML-based format, which is used to exchange security information between different security agents through the Internet. Through the SAML, Web services exchange authentication, authorization, and attribute information. While the XACML is a complement to the SAML and provides a language where a role-based access control mechanism is specified in a declarative format [72].

The XML Encryption and the XML-Digital signature are W3C standard. The XML-Digital signature specifies the XML syntax and processing rules for creating and representing digital signatures which may be applied to the content of one or more resources [77]. Whereas XML Encryption is a standard for encrypting XML elements [78].

Figure 2.6 shows these Web services security standards which come in the quality of service layer of the Web services specification stack [38].

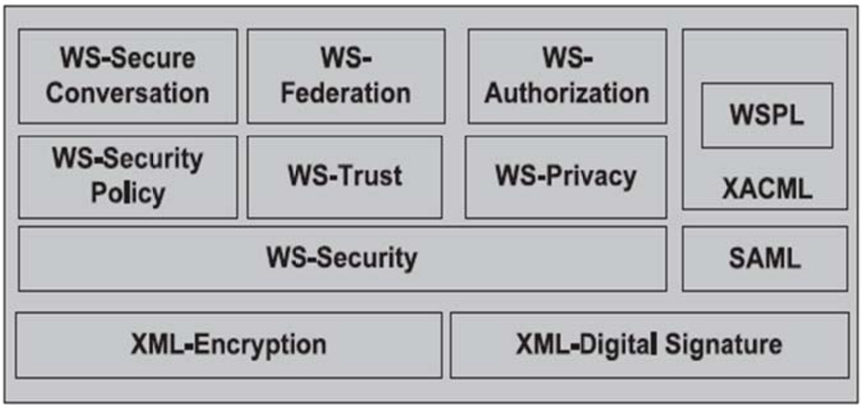


Figure 2.6: Web services Security Standards [38].

2.3 Web Services Composition

The composition of services is a fundamental notion of a service-oriented system. There is plenty of work related to the composition of business services from atomic services [3, 73, 79-83]. Web services are scattered across the Internet, rather then, each service invoking each other by using message exchange patterns, like SOAP, a mechanism is developed to compose more complex interactions among Web services. At the composition layer of the Web service specification stack, the execution order of a service invocation and their interaction patterns are defined. Services are composed using services composition languages/standards like BPEL. A composition consists of the invocation of Web services in the form of Choreography or Orchestration [84]. In *Orchestration* there is a central control which describes the execution order of Web services and the BPEL standard is used for services composition.

While in *Choreography,* there is not a central control and web services interact with each other in a peer-to-peer fashion and the WS-CDL standard is used for services composition. To implement business collaborations, Web services provided by different vendors can be inter-connected, which leads to a composite web service. Composed services are provided by gluing together the WSDL services and corresponding operations [38, 73]. Currently, the business logic of the composite Web service is expressed with the help of a business process modelling language like the UML or the BPMN [23, 24, 57, 73, 85].

2.3.1 Web Services Composition Languages/Standards

Simple interaction among the Web services using standard messages and protocols is not sufficient in the case where business processes are integrated across enterprise boundaries [57, 73]. Real business scenarios involve long-running interactions, transaction management and state-full invocations; they are also often driven by a workflow engine [73]. This raises the need for Web services composition languages that provides the mechanism to fulfill the complexity of business processes execution [57, 73].

Web services composition languages are built directly on top of the WSDL[57]. Two different communities are working for advancement in Web services compositions namely: 1 :) the Business Process Management (BPM) community; and 2 :) Workflow community [24].

2.3.1.1 The BPM Community:

This community has mainly focused on Web service technology and has come up with a multitude of Web services composition standards [24]; the most popular three standards are discussed below.

1. The most popular language for Web services composition is the BPEL4WS (Business Process Execution Language for Web Services) or simply called the BPEL (Business Process Execution Language). The BPEL is built by combining IBM's WSFL and Microsoft's XLANG (it is an XML based extension of the WSDL). The XLANG is a block-structured language while the WSFL is a graph-oriented language [57]. The BPEL is presently a working draft by OASIS. The BPEL is used for the "*Orchestration*" of the Web services [73].

2. The BPML (Business Process Markup Language) is the standard proposed by the BPMI (Business Process Management Initiative). The BPML was originally developed to enable the standard-based management of e-business processes used with the BPMS (Business Process Management System) technology. However it can be applied to a variety of scenarios, including the EAI (Enterprise Application Integration) and Web services composition. The BPML is a specification language committed to

executable business processes [73]. BPML and BPEL4WS are quite similar and are now being merged in OASIS [24].

3. The World Wide Web Consortium (W3C) presented the Web Services Choreography Description Language (WS-CDL). WS-CDL models the peer-to-peer collaboration among participants with different roles using "Choreography" [73]. Other proposals for choreography are HP's Web Service Conversation Language (WSCL) and the SAP/Intalio/Sun/BEA's Web Service Choreography Interface (WSCI) [24].

2.3.1.2 The Workflow Community:

This community is working outside the domain of Web services and is focused on established technologies which are now extended with Web service capabilities. They also support different forms of composition languages. The Workflow Management Coalition (WfMC) provides a specification for interchange of composition models called the XML Process Definition Language (XPDL) [24].

In general, there are many more standards for Web services composition that one can find in literature. The abundance of these overlapping standards is overwhelming. "*In fact, the collection of competing Web services standards without a clear added value has been termed the Web Services Acronym Hell*" [57].

2.4 Business Process Modelling

Business process modelling is gaining more and more attention in organizations because it is the foundation to describe the organizational workflow [20]. Business processes are the key factors for integrating an enterprise; and successful software applications start with the understanding of the business processes of an organization [86]. A business process model facilitates the stakeholders to understand the different aspects of the business system and provide a common platform to discuss and agree on important issues for achieving the business goals [4]. A business process is defined as "*a set of procedures or activities which collectively pursue a business objective or policy or goal*" [4]. It can also be defined as "*a set of activities and execution constraints among these activities*"[20]. For business process representation, different techniques are used; Damij, N. [87] grouped them in two categories; diagrammatic and tabular. This research work focused on the diagrammatic business process representation. Christian Wolter et al. [8] illustrated a few popular diagrammatic business process modelling notions e.g. UML, BPMN, XML Process Definition Language (XPDL), jBPM Process Definition Language (JPDL); among these the UML and BPMN are considered as industry standards for business process modelling [4].

The importance of business process modelling becomes obvious with upcoming of the SOA paradigm. The arbitrary combination of loosely coupled services to give the sense of service-orientation is only possible on the basis of a meaningful and executable model. During the business process modelling the focus lies on the optimal support of the business process whereas IT plays a supporting role in the background which enables the enterprises to cope with market challenges and with new business regulation in a flexible and agile way [54].

A business process could be considered as a special type of service which is built by orchestrating and composing system services; and many standards exist in the software market to support business processes like the BPEL [88].

2.4.1 Security Enhanced Business process Model.

The main focus of the business process model is to model the functional correctness; however, it also focuses on the interacting behavior of the system. Since most security threats are originated at this level, the resulting business process modelling is an important tool for studying and capturing security objectives. Furthermore, empirical studies reveal that common business process experts are able to express their security requirements at the business process modelling level [89, 90].

In the MDS, security is modelled along the business process modelling at the platform-independent model. In this approach functional models are enhanced with security extensions. The MDS paradigm can be used along with a variety of diagrams such as system structure, system collaboration, business process and deployment diagrams. This research work focuses on the business process diagram and uses the UML activity diagram.

For security enhancement two types of descriptions can be used: formal and informal. An informal description is a text-based description written in natural language; whereas, a formal description uses a formal language with a defined syntax. The informal specifications may be problematic because those are written in natural language; hence, the statement may be ambiguous and different model readers may interpret them differently [34]. This research work uses the formal definition of security enhancement, which is easy to define and validate and is known as a high level security requirements.

2.4.2 Business Domain expert is not a security expert.

Usually a business domain expert is involved in modelling the functional diagram and application architecture. It is a fact that he/she is not a security specialist. He/she is familiar with the common security concept but does not have expertise as to how these security concepts are realized. He/she can define a business process model as a UML activity diagram and add security extensions as a UML stereotype [30, 36]. Many researchers like D. Basin [37], M. Hafner [38] and F. Satoh [39] deals with similar processes to realize the security modelling concepts.

A business domain expert does not concentrate on security technologies and security mechanisms that is why the role of a security expert is required. A security expert knows the appropriate concrete security mechanisms and security technologies that can be used to meet the specific security objectives.

2.5 Model Driven Software Development

Currently, software engineering is greatly influenced by the MDE which moves the focus of software development from implementation to the problem domain by raising the abstraction level in the software artifact development and automating implementation with a means of transformation [9]. The primary purpose of the MDSD was "*to alleviate the burden by offering tools and methods to counter the problem at its root: streamlining of the software engineering process, switching to open software architecture and supporting the management of dependencies between components*" [91]. The MDSD is a technology and standards independent methodology based on the concept of a model i.e. it works at model and metamodel levels and is called model-centric [92]. Software systems are specified and developed through models, which will go through the process of stepwise refinement, transformation functions are automatically performed among models at different levels of abstractions as well as among models to code [38, 93]. Different levels of abstractions are presented through models [32, 38]. "*Software has the rare property that it allows us to directly evolve models into full-fledged implementations without changing the engineering medium, tools, or methods*" [94]. Current modelling technologies leveraging this property, have reduced the accidental complexities associated with handcrafting complex software [94]. In this way models become primary development artifacts [40].

The vision of the MDSD is an era of software engineering where modelling completely replaces programming i.e. the systems are entirely generated from high-level models, each one specifying a different view of the same system [27]. The MDSD can be seen as the new generation of visual programming languages which provides methods and tools to streamline the process of software engineering. Productivity of the development process is significantly improved by the MDSD approach and it also increases the quality of the resulting software system [32, 38]. The MDSD is particularly suited for those software applications which require highly specialized technical knowledge due to the involvement of complex technologies and the large number of complex and unmanageable standards like the SOA [38, 55].

2.5.1 Role of Model in MDSD

Models are used to abstract the important aspects of a particular problem [38, 94] and can be defined as "*a set of statements about some system under study*" [95]. However, in the area of software engineering "*a model is an abstract representation of some system structure, function or behavior*" [38]. In the software engineering fields, models have been used for many decades and they are adopted into the software lifecycle. Currently the development methodologies are code-centric[94] i.e. in these methodologies, models are considered as second class development assets, i.e. they are just used for documentation purposes [40] while model based design methodology is being widely accepted in the development of electronics systems due to their flexibility and tool support [43].

In the Model Driven Development paradigm, rather than just a visual aid for communication, documentation and understanding; models are considered as an essential part for the definition of software [38, 40, 55]. They are the cornerstone of software and system development and are used "*to abstract away irrelevant details, rigorously specify the interplay between security and functional requirements, and provide a basis for analysis and transformation*" [27].

After introducing the MDSD and role of model in it, following is a brief description about the main concepts and terms related to the MDSD [38], which are required for the realization of this research work.

2.5.2 Model Driven Architecture (MDA)

The Object Management Group (OMG) launched the MDA as a framework of MDE standards, in 2001 [94]. The idea behind the OMG's MDA framework is to change the focus from technical detail to abstract concepts, i.e. models, which are more stable, more intuitive and would change less [38]. The most important activities in software development through the MDA approach is to model the different aspects of the system and define the transformation rules between the models which will automate the whole development process [58]. The MDA framework advocate to specify three levels of abstraction: Platform Independent Model (PIM), Platform Specific Model (PSM) and Implementation Specific Model (ISM) [38, 94, 96]. UML modelling is used to capture the domain knowledge at the PIM level, which is transformed to either other PIM or to PSM that specifies the intended platform of the system. The PSM is transformed into an ISM which is a runtime environment where the system has to operate. Figure 2.7 illustrates the whole workings of the MDA framework.

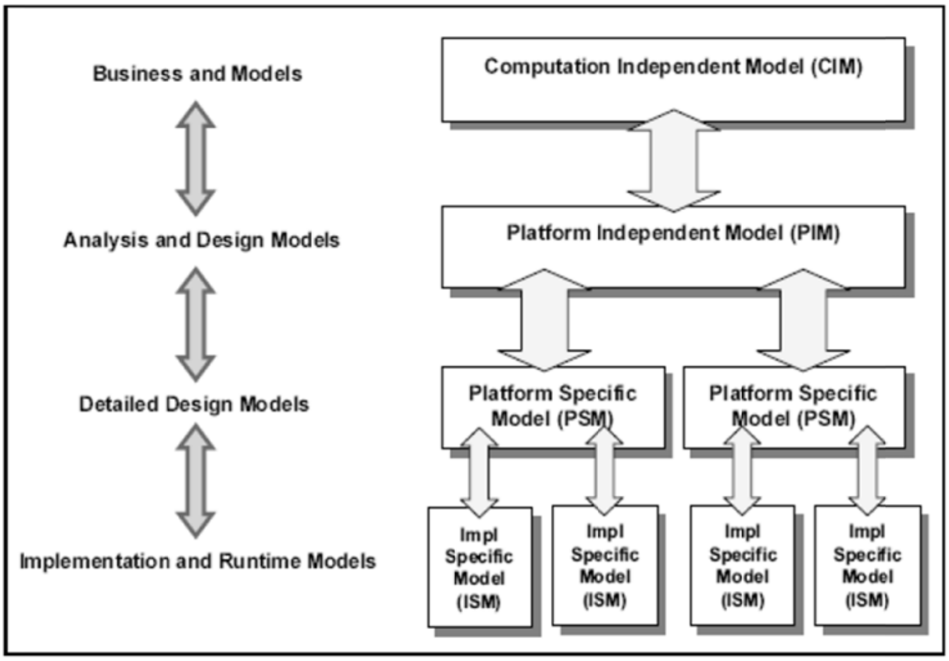

Business and Models
Analysis and Design Models
Detailed Design Models
Implementation and Runtime Models
Computation Independent Model (CIM)
Platform Independent Model (PIM)
Platform Specific Model (PSM)
Platform Specific Model (PSM)
Impl Specific Model (ISM)
Impl Specific Model (ISM)
Impl Specific Model (ISM)
Impl Specific Model (ISM)

Figure 2.7: The OMG's Model Driven Architecture [96]

The main components, also known as pillars of the MDA are the Meta Object Facility (MOF), the UML, and the QVT (Query, View and Transformation). The UML is a general purpose modelling language used to specify, construct and document the artifacts of a system. The MOF is a language used for defining abstract syntax of a modelling language; whereas, the QVT is a standard used to specify and implement the transformation between models, i.e. transformation from a PIM to a PSM [94].

The key to successful application of the MDA is the use of standards [58], that is why the OMG is promoting the MDA approach and related standards like the UML and the MOF because these can reduce the software development cost and improve the quality of software applications [38].

2.5.3 The OMG's Metamodel Architecture.

The OMG's metamodel architecture is used for the definition of the modelling languages. The OMG has defined a four layered architecture namely M0, M1, M2 and M3. Basically, these are the different conceptual levels making up a model: the instance, the model of the system, the modelling language and the metamodel of that modelling language [58]. Figure 2.8 describes the workings of the OMG's metamodel architecture.

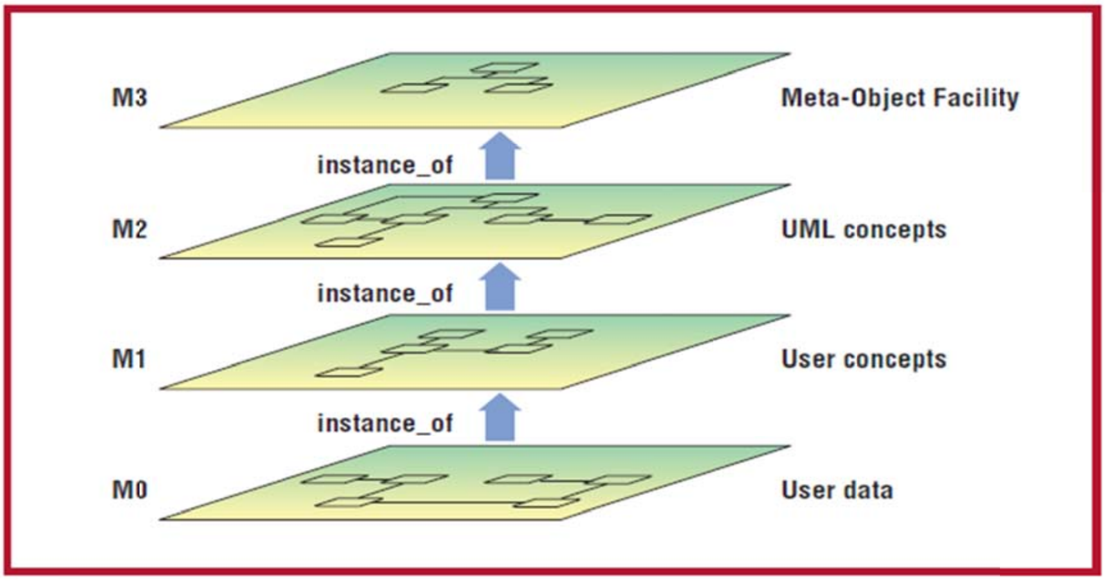


Figure 2.8: The OMG's Metamodel architecture [97]

For defining a new modelling language, metamodelling is performed at the instance at level M2. The metamodel at level M2 itself is an instance of level M3. The instance at level M1 is an instance of the metamodel defined at the M2 level. The instance at level M0 is an instance of the model at M1 level. For example, the modelling language UML is defined and formalized at the M2 level and its metamodelling is performed. Elements of the M2 level itself need to be defined at the M3 Level by using the modelling language MOF [38, 58, 98]. At the M1 level, application modelling is performed using the UML, and a model is created e.g. the UML-Class diagram. At the instance of level M0, instances are created for the application model defined at the M1 level e.g. objects are created for the classes defined at the instance of level M1 e.g. the UML-Object diagram. Figure 2.9 explains the whole phenomenon with an example [99].

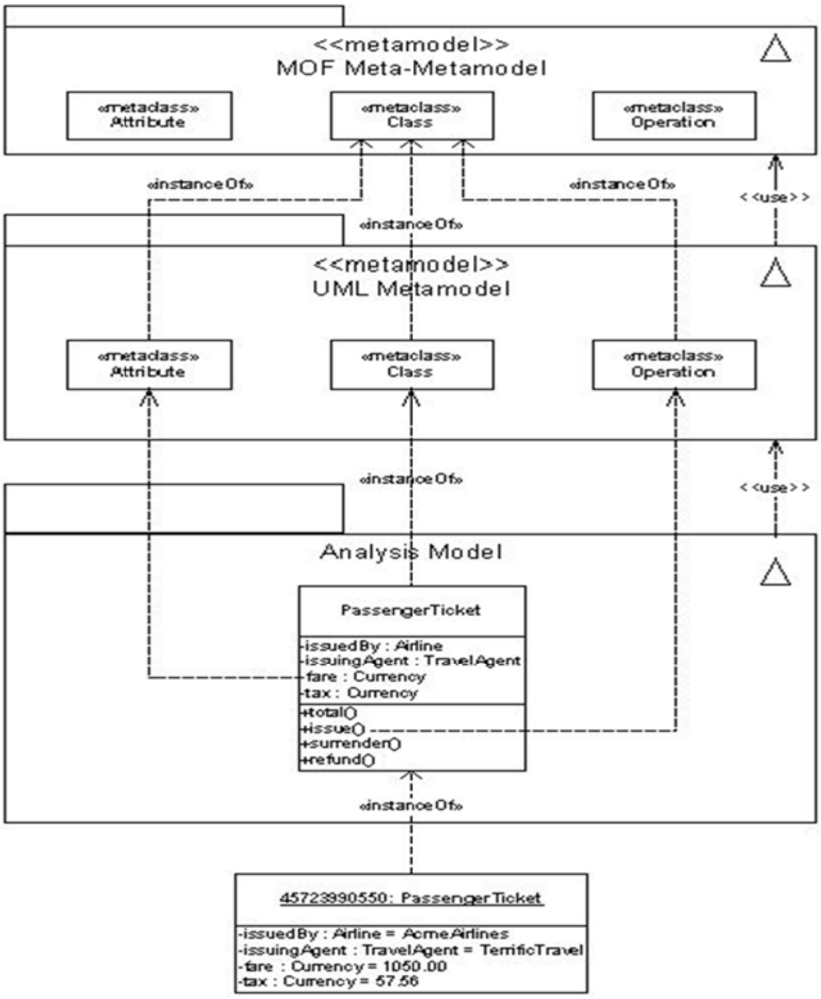


Figure 2.9: The UML Class Diagram Metamodel Architecture [99]

The modelling framework of the MOF is based on standard object oriented concepts and used to formalize the metamodels of a modelling language. It is used to integrate the various models into one “language”. The metadata architecture of the MOF is a reference model for metamodelling [38, 98]. In summary, a modelling language like the UML can be described by its metamodel and the MOF is a modelling language used to describe the metamodel of this modelling language [58]. This layered architecture supports the easy accommodation of new modelling standards as the MOF instances at the M2 level. MOF-aware tools support the manipulation of new modelling standards and enable information interchange across MOF-compatible modelling standards [97]. Proposed Domain Specific Language is defined at the M2 level of the OMG’s Metamodel Architecture.

2.5.4 Metamodel

Metamodelling is a key activity in the MDSD and they are basically models of models i.e. they are used to describe the possible model structure. A metamodel is defined as " *the result of capturing concepts and rules of a specific modelling language via more or less formal means*" [43]. Metamodelling can also be defined as " *the analysis, construction and development of the frames, rules, constraints, models and theories applicable to and useful for modelling a predefined class of problems."[100].* Metamodels are used to describe the abstract syntax, i.e. the concept that needs to be modelled, the context-dependent meaning and the static semantic of a modelling language. Elements of a metamodel itself needs to be defined. In the MDSD, models are taken as an input (source) and generated model or code as an output (target). Transformation from the source model to the target model is defined on the basis of source metamodels and target metamodels. Metamodelling language is used to model the abstract syntax (metamodel) of a modelling language. If a model that is constructed on the basis of a metamodels, respect the modelling rules defined in the metamodel then it is said that this model conforms to the metamodel just like a program conforms to the programming language in which it is written [43]. To organize a landscape of model, metamodelling techniques have emerged; theories and methods are provided for the development of a coordinated representation suitable for heterogeneous environments such as an SOA [43].

2.5.5 Model Driven Security (MDS)

"*Model driven security is an engineering paradigm that specializes Model Driven Software Development towards Information Security*" [38]. The MDS is based on the MDSE and MDA where security requirements are realized at the model level and kept separate from the underlying security architecture. The MDS is an engineering discipline which is concerned with the integration of security requirements in all system development phases e.g. analysis, design, implementation, testing etc. [10]. The vision of the MDS is to provide a way for software engineers to bridge the gap between the system design requirements and security requirements by taking a model-centric approach. This in turn necessitated bridging the gap between security modelling languages and design modelling languages, leading to the notion of security-design modelling languages, such as the SecureUML [27].

In the MDS, security requirements are defined as a model during the designing phase and concrete security configuration files can be generated by the model transformation e.g. security concepts are modelled side by side with the business process modelling at the PIM level of abstraction and step-wise refined to further levels of abstraction i.e. PSM and ISM [30, 38, 53, 55, 93]. Figure 2.10 illustrates the whole process.

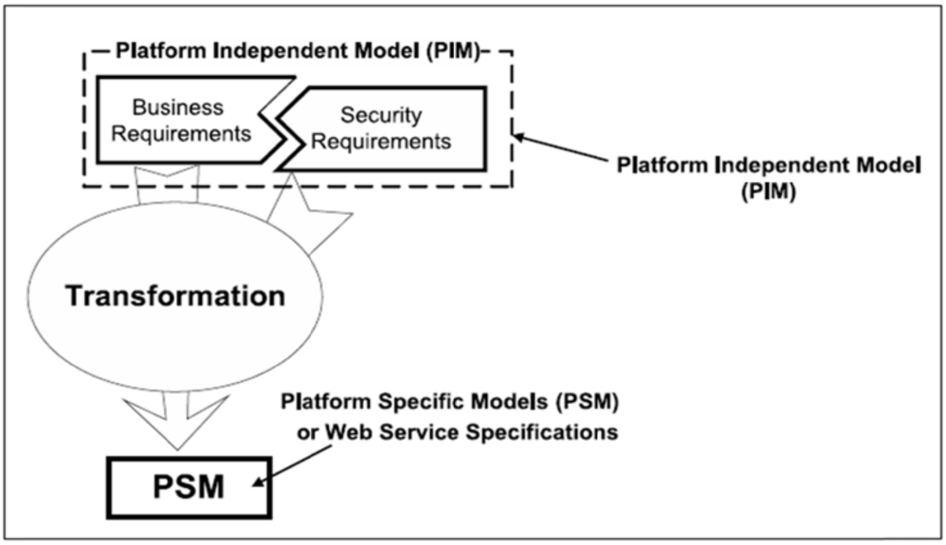

Platform Independent Model (PIM)
Business Requirements
Security Requirements
Platform Independent Model (PIM)
Transformation
Platform Specific Models (PSM) or Web Service Specifications
PSM

Figure 2.10: MDA Pattern with Security Extension [10]

These security objectives are defined in the model with the help of a DSL and transformed into enforceable security rules with a little human intervention. The MDS is a critical component of future Information Assurance (IA) architectures, especially for agile IT environments such as SOA [1].

2.6 Domain Specific Language (DSL)

Application structure, requirements and behavior according to a specific domain are formalized in the form of a DSL which is one of the components of the MDSD. A domain can be defined as " *a field of application delimitated by a specific area of interest*" [38]. A DSL is defined as " *A concise, precise and processable description of a viewpoint, concern or aspect of a system, given in a notation that suits the people who specify that particular viewpoint, concern or aspect.*" [38]. A DSL consists of constructs that capture information regarding the domain it describes[94]. A DSL may also be called a Domain Specific Modelling Language (DSML) [40].

There is a considerable body of accumulated theories and experience to assist programming language designers; however, this is not the case for the designers of modelling languages, which is still an emerging field with very few proven and established guidelines and patterns [101].

2.6.1 Basic Concepts of DSL

A DSL consists of following three concepts [38].

1 *Abstract syntax:* defines the basic concepts, their relationships and the integrity constraints of a DSL[102] e.g. in the OMG's metamodel architecture, the UML Class diagram at the M2 level of abstraction [97]. Normally abstract syntax is defined through a metamodel [38].

2 *Concrete syntax:* defines the notion of the language, which will be used during modelling i.e. the front end of the DSL. These notions may be visual or textual [38]. For example UML notations [97].

3 *Semantic:* of a modelling language defines its meaning in context. Semantics are either defined formally or should at least be documented in an informal way [38]. For example, the natural language specification [97].

2.6.2 Definition Mechanisms/Types of DSL

The specification of a DSL that allows the software products to be represented without ambiguity at a conceptual level is one of the most important concerns when elaborating a Model-Driven development solution [103]. One of the major challenges, an architect of the MDE modelling languages faces is *the abstraction challenge*: *" How can one provide support for creating and manipulating problem-level abstractions as first-class modelling elements in a language?"* [94]. To tackle this challenge, the following two schools of thought have emerged in the MDE community: 1) The Extensible General-Purpose Modelling language School and 2) the Domain Specific Modelling Language School [94, 104, 105].

2.6.2.1 The Extensible General-Purpose Modelling Language School:

Here general-purpose modelling languages like the UML are provided as a base with facilities to extend them with domain specific abstraction [94, 104, 106]. These languages are very successful and they also provide tool support ranging from requirement engineering to code generation [94].

DSL definition: Using Extension Points provided by the Language itself: The easiest way of defining a DSL is the usage of the extension points provided by the language itself [107]. Enough precision for effective MDD processes are not available in the semantic of certain UML conceptual constructs; however, to overcome this limitation, the UML provides semantic extension points defined in the UML specification where different semantics for one conceptual construct or the lake of definitions of appropriate semantic can be found. To introduce the semantic precision into the UML, different extension mechanisms are used; the UML-Profiling extension mechanism is the most popular one. A UML profile describes how UML model elements are extended to support usage in a particular domain [94, 101]. A profile is a lightweight extension mechanism and thus cannot be used to add new model elements or delete existing model elements [94]; to introduce new language primitives, stereotypes are used by extending the semantics of existing model elements present in the UML metamodel [108]. Stereotypes are represented by double angle brackets e.g. <<stereotype>>. The stereotype definition is consists of three things: 1) a user-defined stereotype name, 2); a specification of the base UML concept for the stereotype e.g. Class and some optional constraints that specify how the base concept was specialized for example, a Class that can have at most one parent, and 3) a specification of the semantics that the stereotype adds to the base concept semantics [101]. To formalize the properties of these new language primitives, tagged values are used which are written within curly brackets e.g. {Tagged, Values}, which associate data with model elements. Model elements are assigned to these new language primitives and then they are labelled them with corresponding stereotypes. If some additional restrictions are required on the syntax of these new language primitives, Object Constraint Language (OCL) constraints are used. The OCL is a specification language provided by the UML, based on first order logic. Normally, OCL expressions are used for various purposes such as invariant for classes, pre and post conditions for methods and guards for state diagrams. A set of such definitions i.e. stereotype, tagged values and the OCL constraints constitute the *UML profile*. The UML profile is a specially designed UML package containing a collection of related stereotypes [44, 46, 93, 101]. Most of the current UML modelling tools can readily be used because they support the definition of custom stereotypes and tagged values. Because of having tool support, this approach is widely used [46, 104].

Limitations of the approach:

- UML 2.0 profiling mechanisms do not support the semantics associated with extensions that is why they cannot be used to develop domain specific UML variants that support the formal model manipulation required in an MDE environment [46].
- It is very clumsy to add domain-specific restrictions in large languages like the UML; furthermore for formal analysis, large languages usually lack detailed formal semantics [107].
- Visualization of the complicated security objectives might be confusing; furthermore, many modelling languages do not provide extension points [46].

2.6.2.2 The Domain Specific Modelling Language School:

In this type of language extension mechanism, a DSL is provided using the OMG's MOF meta-modelling mechanism [94, 104]. DSLs are small and provide a basis for domain-specific formal analysis and use those notions which are familiar to domain experts [46]. The DSL is used to formalize a modelling language capable of formalizing different business domains (like e-government, e-health, e-education), system aspects (like security, real-time) or concrete technologies(such as EJB or .NET) [93]. These extension techniques are *metamodel based* techniques and known as heavy weight extension mechanisms. The metamodel based technique of defining the DSL is mostly used when the "*domain*" is well defined and has an accepted set of concepts; there is no need to combine the domain with other domains and the model defined under the domain is not transferred into other domains [93]. Under this school of thought, there are two types of extension mechanisms:

A. *DSL Definition: By extending the Metamodel of Existing Modelling Languages:*

The DSL can be defined by using the MOF by extending the metamodel of an existing modelling language e.g. UML. The abstract syntax of the DSL is represented by the metamodel and notions (concrete syntax) are specified with the UML profile [43, 46, 93]. In this way, an existing metamodel is reused and specialized. Stereotypes are used to formally extend the metamodel of an existing modelling language, and at the modelling level, stereotypes are manipulated as annotation on the model elements.

Limitations of the approach:

- Extensions are defined and integrated according to a particular domain into a specific modelling language based on its metamodel [58].
- The extended and customized metamodel is based on the entire metamodel of existing modelling languages and may be complex [94].
- During this approach, manual changes are applied to the metamodel of an existing modelling language which is tedious and error prone due to many reasons: 1) difficulty in ensuring that the changes are made consistently across the metamodel, 2) difficulty in determining the impact of the change on other model elements, and 3) difficulty to ensure that the resulting modified metamodel is complete and sound [43, 46, 93].
- To support the DSL; a CASE (Computer Aided Software Engineering) tool may also require extension to accommodate these new language primitives in a particular storage component (repository) and visualization component [107].

With all their limitations, unless there is a real need to deviate from the UML's metamodel, the benefits of using the UML Profiles undoubtedly outweigh their limitations [43]. This DSL definition approach is used in the proposed work.

B. DSL Definition: By Defining a New Metamodel having no dependency on existing Modelling Languages:

A new DSL for modelling the domain of interest or a particular problem is created by a fully dedicated metamodel using an MOF having no dependency on existing modelling languages. The resulting DSL has a much more concise vocabulary than the vocabulary of general purpose modelling languages e.g. the UML or BPMN. For querying and manipulating meta-data of these DSLs, the interface would be simpler then the UML interfaces. This way of extension is optimally suited for specific problems at hand [58]. An example of such a language is the Common Warehouse Metamodel (CWM) [46].

Limitations of the approach:

- Sometimes it does not provide the well-defined mapping between the UML model with which developers work, to the instances of the metamodel of the DSL that defines the meaning of this model [58].
- Tool development to support the DSL is a very difficult and expensive task due to the sophisticated semantics behind the modelling language construct [101].

2.6.3 Current practice of defining DSLs

There is no universal approach for the integration of security and design modelling languages [107, 109]. The current practice of defining a DSL by different researchers in the related work [4, 8, 44, 56, 93, 101] is that the abstract syntax of the DSL is represented by a metamodel and the concrete syntax is represented by a UML Profile.

This work is also working along the same approach and defines the abstract syntax of the proposed DSL by a metamodel; and the concrete syntax by a UML Profile.

2.6.4 Selection of a modeling language

UML and BPMN are considered as industry standards for business process modelling [4]. Below a brief description of both languages is provided followed by the comments why UML is selected for this research work.

The BPMN 1.0 specification is proposed in May 2004, by the Business Process Management Initiative (BPMI.org). The OMG has adopted it for standardization purposes in February, 2006. The primary goal of the BPMN is to make business process modelling easier by reducing the gap between technical and business people. BPMN was designed for modeling business process and has a primary goal of being understandable by all business stakeholders [110]. BPMN only support the concepts of modelling a business processes and it do not support modelling the other concepts of software systems [111].

The UML was published by the Object Management Group in 1997 [112] and it is currently upgraded to version 2.0. The scope of the UML is very broad and it covers a large and diverse set of application domains. UML 2.0 provides around thirteen different kinds of diagrams for modelling different aspects of a software system [113]. UML Activity Diagram is used for business process modeling [114].

Currently the Proposed DSL "UML-SOA-Sec" is used to model the security along the business process modelling. However the scope of UML-SOA-Sec is not narrow, once a DSL is formalized, it can also be used to model the other aspects of software systems. That's why UML is the best choice because it covers almost all the aspects of a software system.

2.6.5 Unified Modelling Language (UML)

The OMG has provided many modelling languages and among them the UML is the most widely used and accepted language [58]. The UML is a general purpose modelling language that can be applied to different application domains (e.g. healthcare, telecom, banking etc.) and different implementation platforms (e.g. .NET, J2EE, CORBA etc.) [58].

The UML has many features which motivate its selection for this research work and they are summarized as follow: [38, 94, 115]

- The UML is the industry's de-facto standard for software modelling.
- It is a graphical language for modelling object-oriented systems.
- It separates abstract syntax and concrete syntax.
- It provides extension mechanism through a profiling mechanism.
- It raises the level of abstraction.
- It is platform independent.
- The Object Constraint Language (OCL) is tightly integrated.
- Tools are available to model standards and to define profiles for various specific modelling purposes.

The UML is formally proven as a visual language for modelling object oriented systems and it provides different diagrams to represent the different structural and behavioral aspects of a software system [10].

2.6.6 UML Profile

The Profile packages included in the UML 2.0 defines a set of UML artifacts that allows the specification of an MOF model to deal with specific domains (e.g. Business Process Modelling, Finance etc.) or implementation technologies (e.g. .NET, J2EE etc.) [58].

The UML 2.0 outlines several reasons to customize a metamodel [58]:

- To have a terminology that is adapted to a particular platform or domain.
- To have syntax for constructs that does not have a notation.
- To have a different notation for already existing symbols, more appropriate for the target application domain.
- To add semantics left unspecified in the metamodel.
- To add semantics that do not exist in the metamodel.
- To add constraints that restrict the way you can use the metamodel and its constructs.
- To add information that can be used when transforming one model to another model or to code.

A UML Profile is a set of the above mentioned extension mechanisms, grouped into a UML package known as <<profile>>. The extension mechanism can extend the syntax and semantics of UML elements; however, it must respect the original semantics of these UML elements, i.e. the UML profile cannot change the semantics of the UML elements. Several UML profiles have been adopted and standardized by the OMG and are available for the public use. The number of the OMG's UML Profiles is rapidly growing [58].

The following are guidelines for defining a UML Profile for a particular application domain [58]:

1. First of all, a set of elements that comprise the particular domain needs to be defined and the relationship among these elements needs to be expressed in the form of a metamodel.
2. For each element of the metamodel, which we want to include in the UML Profile, a separate stereotype is defined inside the <<profile>> package. In order to clarify the relationship between the elements of the metamodel and the

stereotypes of the profile, names of the stereotypes of the profile are assigned according to the elements of the metamodel.

3. Stereotypes of the profile will be applied to specific UML metaclasses.
4. Attributes may be assigned to the elements of the metamodel, which appear in the form of tagged values.
5. Necessary constraints may be applied on the Profile in the form of OCL constraints.

2.7 SOA Security

Generally security is considered as a state of freedom from risk or danger; however, in computer sciences it is a field which deals with the risks, threats and mechanisms to the use of a computing system [38]. Computer security is defined as "*A computer is secure if you can depend on it and its software to behave as you expect*" [116]. However, security is not just like a state only; it also describes other things e.g. the measures to preserve this state. A computer's security can also be defined as [117] "*Computer security deals with the techniques employed to maintain security within a computer system*". These two definitions for computer security can be true for the isolated host; however, they may fall short in the modern computing system e.g. an SOA environment, where loosely coupled components distributed over the Internet are connected. Computer systems are no longer conceived of as a centralized architecture. A system which is connected to other systems is exposed to many additional security threats. That's why a comprehensive security definition is required which also covers the environment to which the system belongs. A very comprehensive security definition for an SOA system is "*the sum of all techniques, methods, procedures and activities employed to maintain an ideal state specified through a set of rules of what is authorized and what is not in a heterogeneous, decentralized, and inter-connected computing system*" [38].

2.7.1 Challenges of SOA Security

In an SOA environment, software applications are not considered as an isolated host. Many partners are working together to achieve a business goal and they span over multiple security domains. These partners may not know each other and want to have control over their portion of the workflow. Business applications seem to be virtual-organizations making a decentralized architecture of a peer-to-peer style [38].

Following is a brief illustration of a few prominent SOA security challenges which help in understanding how security in an SOA is different from other architectural environments.

- In an SOA environment, data after originating from the originator has to travel through multiple intermediates before reaching its desired recipient. Hence, only a secure connection between the originator and recipient is not sufficient to ensure confidentiality, integrity and availability as in the case of traditional point-to-point architecture. Therefore, SOA applications require additional security components and adoption of new security standards and specifications [118].

- SOA applications are composed of different services from multiple venders. Different client applications invoke services in different contexts this means it can never tell how the security would be handled. Applications alone can no longer be in charge of security and security models cannot be hard-coded into applications [119].

- As applications and organizational boundaries are no longer impediments to reuse, traditional approaches of security are no longer suffice [119].

- Human security administrators are not able to define all fine grained security rules with sufficient assurance, to distribute them to all IT systems and to check many log files or admin consoles [1].

2.8 Security Objectives

Security is an abstract concept which can be defined precisely by specifying the set of security goals or objectives [8]. Security objectives describe the most basic security need of an asset [38] and they can be defined as " *a statement of intent to counter identified threats and/or satisfy identified organizational security policies and assumptions*" [120]. These security goals can be further subdivided, specialized or combined [8]. Many names/terms can be found in literature for security objectives like security properties, security aspects, security concerns, security intents or security states etc. [121]. To maintain consistency with in the dissertation, the term security objectives is used

There are numerous SOA security objectives that may differ for each stakeholder like vendors, security experts, consultants, business process experts etc. Similarly security objectives can be about some specific business case, technology, governance, deployment etc. Unclear security objectives result in unclear security implications, which is cited as one of the most important issues that limit the SOA benefits and hence slow down its adoption [1].

Ramarao Kanneganti et al. [119] classified the security objectives into two groups, *functional* and *non-functional*. Functional objectives of security are authentication, authorization, confidentiality, integrity, protection against attack and privacy. Functional security objectives of SOA applications are the same as of traditional software applications. Non-functional objectives of security are interoperability, manageability and ease of development. In this research work, the focus is on the functional security objectives of the SOA systems.

2.9 Security Objectives in Related Work

During this section, a thorough discussion is provided about the security objectives presented in literature. First, that work is presented where authors do not mention anything about the target architecture i.e. either it is an SOA or other architecture, these security objectives are named as "*Security Objectives General*". Afterwards, a detailed discussion is presented about those security objectives where authors explicitly mentioned the SOA environment, these security objectives are named as "*Security objectives SOA*".

2.9.1 Security Objectives General

These are the security objectives irrespective of the deployment environment i.e. it may be considered for any of the deployment environments. Here authors did not mention anything about the deployment environment.

N. Nangaratnam et al. [122] specified the five security objectives for a business process model, namely audit, authenticate, authorize, confidentiality and integrity.

Firesmith [2] has a very comprehensive discussion about the general security of a software application and identified eleven security objectives. The eleven security objectives are: identification, authentication, authorization, immunity, integrity, intrusion detection, non-repudiation, privacy, security auditing, survivability and physical protection.

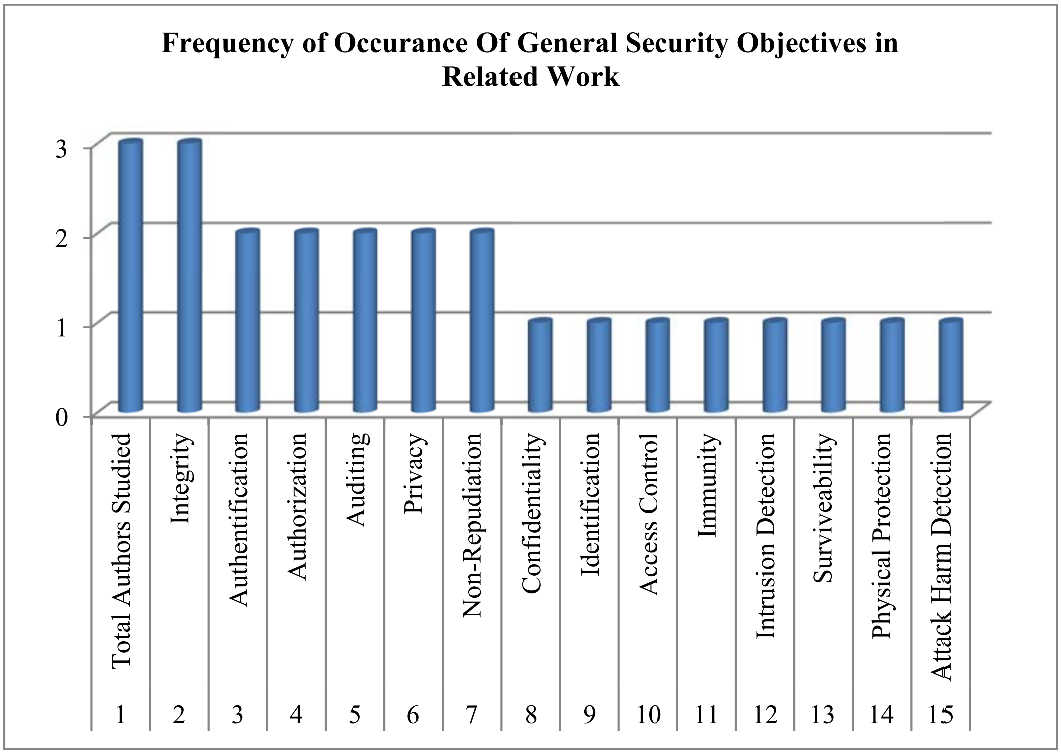
Alfonso Rodríguez et al. [4, 56] focused on five security objectives: access control, integrity, privacy, attack-harm detection and non-repudiation.

The whole discussion is summarized in Table 2.1, representing the general security objectives in related work and Table 2.2, represents the frequency of their occurrence in graphical form.

Table 2.1: General security objectives

S/No	**Security Objectives**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
	Research Groups	**Integrity**	**Authentication**	**Authorization**	**Auditing**	**Privacy**	**Non-Repudiation**	**Confidentiality**	**Identification**	**Access Control**	**Immunity**	**Intrusion Detection**	**Survivability**	**Physical Protection**	**Attack Harm Detection**
1	Nangaratnam et al. [122]	✓	✓	✓	✓			✓							
2	Firesmith [2]	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	
3	Alfonso Rodríguez et al. [4, 56]	✓				✓	✓			✓					✓

Table 2.2: General security objectives in graphical form



2.9.2 Security Objectives For SOA

In an SOA environment, two typical viewpoints exist for security objectives namely *information based* [123] and *service based* [124]. In the information based viewpoint, focus is on information security which has two states; stored information and information in transmission.[6]. Stored information mostly focuses on security objectives of access control i.e. authentication and authorization. While information in transmission focuses on the message-level security, which is ensured by the Secure Socket Layer (SSL) [52]. The second viewpoint, i.e. service based, analyzes the SOA security from two different views; they are the individual service security and services composition security. It defines the security objectives at the service level [6].

Most of the security objectives have been known before SOA; however, now they are simply becoming more evident or even amplified. Security challenges for SOA environment may be different due to its architectural style. After providing a brief idea about SOA security, a detailed literature review is provided about the security objectives focused on by the different research groups for defining their DSL for the SOA applications [4, 8, 12, 38, 56].

Michal Hafner et al. [38] defined three security objectives namely confidentiality, integrity and availability. They defined access control under the umbrella of confidentiality; and availability is used in the meaning of no-repudiation.

Christian Wolter et al. [8] presented a security policy model by focusing on six security objectives: authentication, authorization, confidentiality, integrity, availability and auditing. Michal Menzel et al. [20] used the same security policy model specified by Christian Wolter et al. and defined security extensions to the BPMN.

Michal Menzel et al. [125] specified four security objectives necessary for the SOA architecture: authorization, authentication, integrity and confidentiality.

Yuichi Nakamura et al. [53] defined three security objectives for their work: authentication, integrity and confidentiality and defined a UML profile. In another work Yuichi Nakamura et al. [12] addressed four business level security objectives as they are easy to be understood by the business user namely: authentication, integrity, non-repudiation and confidentiality.

Simon Johnston [126] described seven security objectives which are essential for the SOA environment: identification, authentication, authorization, privacy, auditing, data integrity, non-repudiation.

Ulrich Lang and Rudolf Schreiner [1] describe the five security objectives in their work namely confidentiality, integrity, availability, auditing and manageability.

Thomas Erl [127] presented an overview of the security objectives for the WS Security and presented a framework containing five security objectives namely identification, authentication, authorization, integrity and confidentiality.

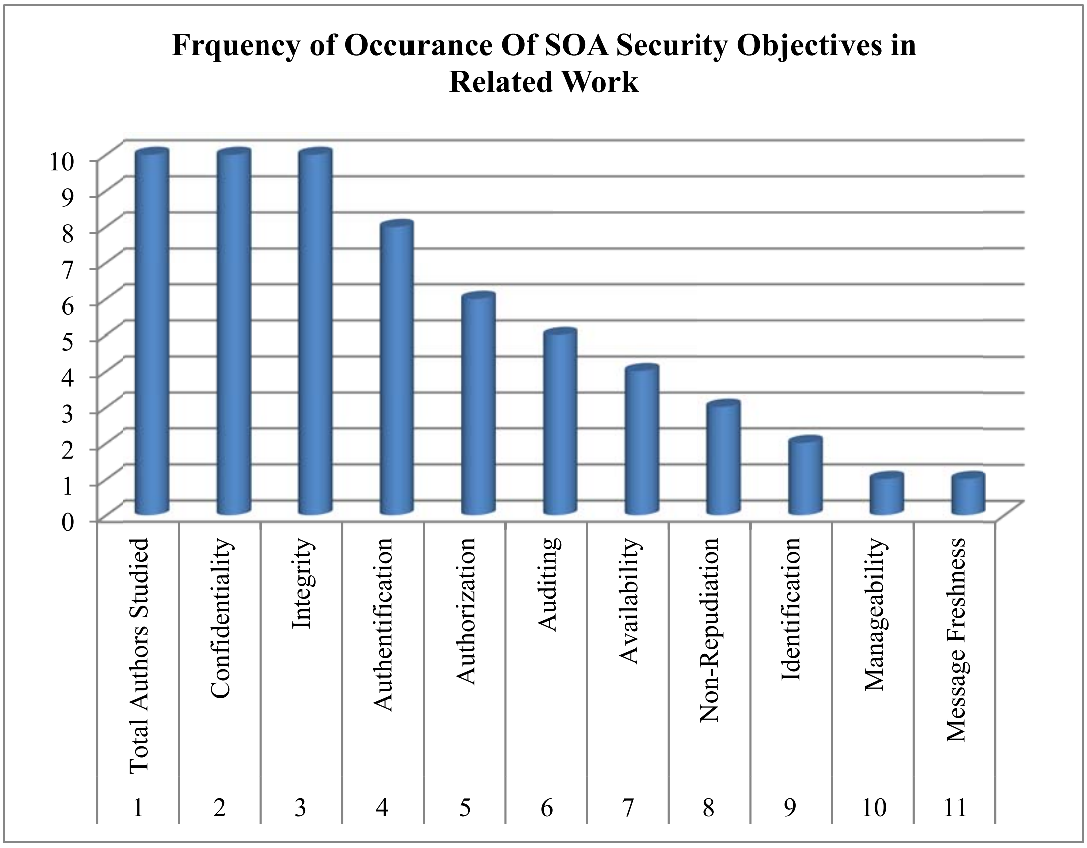
Tan Phan et al. [128] introduced a method for the design and implementation of SOA Business Security Engineering. The security objectives they focused on in their work are confidentiality, integrity, non-repudiation, auditing, authentication, authorization and message freshness.

The whole discussion is summarized in Table 2.3, representing the security objectives for the SOA environment in related work and Table 2.4 represents the frequency of their occurrence in graphical form.

Table 2.3: Security objectives for the SOA environment

S/ No	Security Objectives	1	2	3	4	5	6	7	8	9	10
	Research Groups	**Confidentiality**	**Integrity**	**Authentication**	**Authorization**	**Auditing**	**Availability**	**Non-Repudiation**	**Identification**	**Manageability**	**Message Freshness**
1	Michal Hafner et al. [38]	✓	✓				✓				
2	Christian Wolter et al. [8]	✓	✓	✓	✓	✓	✓				
3	Michal Menzel et al. [20]	✓	✓	✓	✓	✓	✓				
4	Michal Menzel et al. [125]	✓	✓	✓	✓						
5	Yuichi Nakamura et al. [53]	✓	✓	✓							
6	Yuichi Nakamura et al. [12]	✓	✓	✓				✓			
7	Simon Johnston [126]	✓	✓	✓	✓	✓		✓	✓		
8	Ulrich Lang and Rudolf Schreiner [1]	✓	✓			✓	✓			✓	
9	Thomas Erl [127]	✓	✓	✓	✓				✓		
10	Tan Phan et al. [128]	✓	✓	✓	✓	✓		✓			✓

Table 2.4: SOA security objectives in graphical form



2.10 Essential Security Objectives for SOA Application

The focus of this research work is security modelling along the business process modelling for SOA applications; that's why only those security objectives mentioned in the literature are focused, which are to be modelled during the business process modelling Massive literature can be found on software security objectives, however only essential security objectives are selected which can be modelled in the business process model by the business process expert and can be used for the identification of specific security implementation for SOA applications [126]. Although there has been a lot of research on security objectives and concerning technologies, however only business-level security objectives are addressed which are easily understandable for a business process expert [12], who is not well versed with the technical security details, however he/she is able to model it at very abstract level [126].

In the following sub-section, there is a detailed discussion to summarize the work founded in literature about the security objectives, in order to find out the essential security objectives for the SOA application.

2.10.1 Discussion: Finding the Essential Security Objectives for SOA Applications

Computer security is also defined as "*the protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources.*" [129]. These security objectives are applicable to all information systems irrespective of their technology platforms, communication channels, size of the organization etc. Security is a composite notion, comprised of , *confidentiality* (the prevention of unauthorized disclosure of information), *integrity* (the prevention of unauthorized amendments to or deletion of information) and *availability* (the prevention of unauthorized withholding of information) [130]. Conceptually, the three basic security objectives are Confidentiality, Integrity and Availability [118] also known as CIA (Confidentiality, Integrity and Availability) [131].

CIA are termed as pervasive in nature and fundamental to all information systems [132] and for SOA applications, these basic security objectives are unchanged; however, these are not sufficient for the creation of a secure SOA environment [131].

Among the eleven security objectives discussed by Firesmith [2], four are out of the scope of this research work namely; physical protection, intrusion detection, survivability and immunity; this is because proposed work only focuses on those security objectives which are essential to be modelled along the business process modelling for SOA applications. Every organization is supposed to have some security measures for physical protection, survivability, immunity etc.

"*Identification*" is specified as a separate security objective by [2] and [126]. Identification and authentication are assumed when one is trying to model "W*ho are you?*" If the example of an ATM with drawl is discussed, then the ATM card is a bank-issued identification; whereas, the PIN-Code allows the ATM to authenticate the person as an account holder. It is far more important to model the notion of authentication than identification [126]. During proposed work, identification is treated as a part of the authentication process and there is no need to model it explicitly in a business process diagram.

The "*Attack-harm-detection*" specified by Alfonso Rodríguez et al. [4, 56] is a security mechanism which will allow an application to detect, register and notify an attack attempt or a successful attack. It is the same kind of mechanism as described by Firesmith in [2] under the name of "*immunity*" security objective. Every organization is supposed to have security protection mechanisms such as an anti-virus or a firewall; therefore, there is no need to model it explicitly in a business process diagram.

Ulrich Lang and Rudolf Schreiner [1] take "*accountability*" in the sense of "auditing". They also mentioned a security intent "*manageability*" i.e. IT Security should be manageable. Basically it is a concept of overall security related to the SOA environment; therefore, there is no need to model it explicitly in a business process diagram.

Among the fifteen security objectives discussed in Section 2.9 of chapter 2, the following are the essential security objectives, which are focused by different authors either as it is or with some different name or by merging them. Few security objectives are further specialized e.g. confidentiality is achieved through access control; while access control itself is implemented through authentication and authorization mechanisms. Same is the case for Integrity and Availability; they are also achieved through access control. That's why this work divides the security objectives found in the literature into two groups: 1) Security Objectives and 2) Security Mechanisms; security objectives define the basic security objectives while security mechanisms specify how these objectives are realized [38].

2.10.2 Security Objectives

The following are the essential security objectives, which business domain experts will model along the business process modelling.

2.10.2.1 Confidentiality

This security objective specifies the system's state where only authorized entities can access the information [38]. Confidentiality is defined as *"It provides protection* *against the unauthorized notice of stored, processed, or transferred information"* [8, 20]. To represent the data confidentiality security objective, some authors use the term "*Privacy*" [2, 4, 126] and some represent it with the term of "*secrecy*" [125]. In SOA applications confidentiality is enabled through access control (authentication and authorization) [118].

The typical objectives of the data confidentiality are to ensure that [2]:

- Unauthorized individuals and programs do not gain access to sensitive data and communications.
- Access to data and communications is provided on a "*need to know*" basis.

2.10.2.2 Integrity

This security objective identifies an authorized subject to alter information in authorized ways [38]. It ensures the integrity of data (properness, intactness, correctness and completeness of information) as well as the integrity of origin [20, 38, 125]. It ensure that the transferred, processed or stored data can only be modified with proper rights [8]. Basically it ensures that the transferred data between parties must be guaranteed to reach the recipient in the same form and with the same content [12]. The typical objective of the data integrity is to make the data and communication trust worthy [2]. In SOA applications integrity is enabled through access control (authentication and authorization) [118].

2.10.2.3 Availability

This security objective ensures that the data, resources and services which are needed for the proper functioning of a system are available at each point in time regarding the requested quality of service [8, 20]. In an SOA application availability is taken in a sense to minimize threats and vulnerabilities to maintain normal system operations [118].

2.10.2.4 Auditing

It is a process of verification for all actions performed in an information processing system [8]. Basically auditing is performed to verify all operations in an information system [8]. It underlies with each security requirement and will automatically be understood when a security requirement is specified in a model [4].

The typical objective of traceability and auditing security objectives is to ensure that the software application will collect, analyse and report information about the status (e.g. enabled vs. disabled etc.) and use (e.g. change in properties) of its security mechanism [2].

2.10.2.5 Non-Repudiation

This security objective ensure that a user may use a resource or call a service and this usage or service call must not be deniable [38].

Typical objectives of the non-repudiation security objectives are [2]:

- Proper temper-proof record keeping is performing regarding the interactions of the parties to prevent them from denying that it has taken place
- To minimize any potential future legal and liability problems that might be caused due to someone denying one of their interactions.

2.10.3 Security Mechanisms

The following are the security mechanisms through which security objectives are realized.

2.10.3.1 Authentication

It is a mechanism to verify the identity of an entity [38]. It ensures the credibility of information by confirming them as authentic [8, 20]. It establishes the trust relationship between a subject and a party that relies on a claim stated by the subject [125]. The typical objective of the authentication is to ensure that "*externals are actually who or what they claim to be and thereby avoid compromising security with an impostor*" [2]

2.10.3.2 Authorization

Authorization is a mechanism based on some specific security model, in regards to the means to grant various privileges to various entities on different resources [38]. Basically, it is a process of granting rights to participants to perform an interaction or task [8, 20]. It determines the rights which will be granted to the subject based on the trust relationship and properties of the subject's identity [125].

The typical objectives of authorization are to ensure that [2]:

- A person (Administrator of the system) is able to authorize specific authenticated users and client applications to access specific application or component capabilities or information.
- Authenticated externals (users or client applications) can access a specific application or component or information if and only if they have been explicitly authorized to do so by a properly appointed person(s).

2.11 Related Work

The related work is classified into two sections. The first section, illustrates a detailed discussion regarding the related work of the different security extensions presented in the modelling languages from different researchers by presenting DSLs. The second section presents the related work about the different web services composition frameworks which have been presented by different researchers for web services composition.

2.11.1 Definition of Security at Early software Development Phases

General purpose modelling languages like the UML or BPMN; do not have the capability of modelling the security objectives along the modelling of the software system. To model the security objectives related to different system's aspects, different security extensions are proposed by several authors. There is plenty of interesting work and among them a few of the important related approaches are discussed below.

Basin David et al. [32, 93] for the first time introduced the term Model Driven Security. They have presented the "*SecureUML*" to model the security objectives for modelling the static structures of a system. Basically it is a separate language based on the Role Based Access Control (RBAC) protocol and its focus is to generate Access Control Policies from Abstract Authorization Constraints. They have presented a metamodel for an abstract syntax and used the UML profile for the concrete syntax; security constraints are added through the OCL. The approach is flexible and afterwards, the SecureUML provides a schema to create other languages addressing different security aspects. Instead of adopting one-language-fit-for-all, they have proposed a general schema for integrating security objectives into system design models. In their work, they combined the *SecureUML* with a design modelling language based on class diagrams, known as the *ComponenetUML*, and later on with a language based on a state diagram, known as the *ControllerUML* [37]. Afterwards, they combined the *SecureUML* with the language for modelling Graphical User Interfaces and gave it the name *ActionGUI* [133]. They have considered two phases for the Model Driven Security: 1) Definition of abstract access control policies; and 2) Transformation to J2EE Deployment Descriptor Configuration. They have extended the system model with security stereotypes, which means that the domain expert must have knowledge of the security patterns to be used in a particular access control scenario. As domain expert is not a security expert and does not have much knowledge of security mechanisms and security patterns. He should only define the security objectives at a very high abstraction level and then afterwards a security expert or architectural team should refine and transform the model and perform the code generation.

Jan Jürjens et al. [134, 135] have extended the UML and presented a UML profile for the modelling of safety critical systems named the *"UML-Sec"*. Their main idea is that the aspect of security should be considered throughout the whole system development process. They have used different UML models at various levels to capture security objectives like the UML Class diagram (defining security for class attributes and functions), the UML Sequence diagram (for defining message security to exchange cryptographic data) and the UML Deployment diagram (defining security for physical components). Their work mostly focused on the formal definition and validation of security by the developer, who already has knowledge of security; however, they did not address how business department will address security.

Michael Hafner and Ruth Breu have worked along the area of work-flow security and have proposed a MDS framework known as SECTET [28, 38, 136]. They have presented the SECTET-DSL (a DSL used to model the inter-organizational workflows) and SECTET-PL (a policy language used to define the abstract security policies). For the SECTET-DSL; they have presented security stereotypes in a UML activity diagram. In the SECTET framework, modelling is performed to represent two kinds of views: workflow-view and interface-view. Basically in their work they are focusing work-flow security. M. Alam [10, 55, 137, 138] has also worked along the same direction and presented the Role Based Access Control (RBAC) policy for a distributed system. M. Memon [29, 139] has presented an enhanced form of the SECTET framework and named it the SECTTISSIMO framework. In the SECTTISSIMO framework, after the PIM, a new layer is added namely the Abstract Security Service Model to further elaborate the security objectives. He has extended the SECTET-DSL and used it in his framework. In their approach, the abstract security policy is directly converted into a code requiring the domain expert to have enough expertise to incorporate a security pattern at the early stages of the system development.

Rodriguez A. et al. have created a metamodel for their security extensions and defined the security stereotypes and also assigned different symbols to these security stereotypes. They used the same metamodel and security stereotypes for extending both the popular modelling languages, i.e. the BPMN [4] as well as the UML [56]. They are working along the area of business process modelling. Most of their work remains at the descriptive level and they only model the system with a security annotation. Later on Rodríguez A. et al.[92] proposed the generation of use-case views out of business process models which are examined for security requirements. They are defining security in term of general security and using the MDA approach to define security at the PIM level of abstraction. In their work, they do not specify the target architecture i.e. whether it is centralized, distributed or an SOA etc. Their approach is closely related to our work except we are focusing on the specification of security stereotypes for an SOA environment which results in focusing on different security objectives.

Christian Wolter et al. [8, 30] have introduced high-level security policies for confidentiality, integrity, authentication, authorization, availability and audit. For each security objective, they have presented a generic security policy model, which captures the relations between basic entities like objects, attributes, their interactions and the effects of these interactions. The model includes views on the enterprise architectural space which allows connecting elements from different perspectives. The security policy model maps the security goals to the security constraint model, which are elaborated in the next phases. Michael Menzel et al. have worked along the same direction [33]; they have presented a metamodel for the model-driven generation of security policies for the SOA system. Their metamodel describes the basic entities, their relations and associated roles (such as service and service consumer) in an SOA environment and provides the foundation to model interactions and the exchange of information. They have introduced the security constraints on the security policies described by Christian Wolter et al. [30]. In [20] they proposed an approach to describe security objectives at the business process layer and their translation to concrete security configuration for the SOA based system. They introduced security objectives for business process modelling such as authentication, authorization, trust, data integrity and data confidentiality, system integrity and system availability; these are to be modelled in a business process model. These security objectives evaluate the trustworthiness of participants based on a rating of enterprise assets. Later on they tried to address the problems of security in services composition by providing a solution based on modelling concepts, semantic technologies and trust levels to express manage and negotiate security requirements in a technology-independent way [36]. In their work, they have mentioned the security pattern; however, did not define how these patterns would be selected and used. In [107] they defined a modelling language for modelling security at the system design level for SOA applications and named it the "SecureSOA". It is an extension to the SecureUML by David Basin et al. [32, 93] for the service-based systems. They are not focusing on any specific diagram rather discussing Fundamental Modelling Concepts (FMC) which can be used to model the structure of a system, processes in a system, and value domains of a system. Their work regarding the services composition is just a guideline for the composition of services.

SOA environment focus the security objectives related to the information security [123] and service security [124]. After combining these two viewpoints; Siming Kou et al. [6] presented another viewpoint called the organizational view and also added more security objectives, which are related to an organization e.g. trust, auditability and system level reliability. They have presented three different metamodels to present the three different viewpoints for modelling different security objectives of an SOA based application. These metamodels have extended the SoaML (Service oriented architecture Modelling Language) with security viewpoints to support the Model-Driven

Engineering approaches for the design and development of SOA based systems. The SoaML specification is a UML profile and a metamodel for the design of services for a service-oriented architecture [140]. The name of their proposed language is SoaML4Security (SOA Modeling Language for Security). As focus of this research work is security modelling along business process modelling for SOA system; that is why proposed work focuses on only those security objectives, which are to be modelled during the business process modelling.

Bertino et al. [141] have presented how the WS-BPEL is enriched by authorization information for access control and introduced the Business Process Constraint Language (BPCL) which allows formulating the authorization constraints. A WS-BPEL engine has been extended to be able to interpret these access control constraints. The BPCL is limited to users, roles and activities i.e. the Role Based Access Control (RBAC). Due to its technical focus, it is not adequate for use in a business department.

F. Satoh et al. [39, 53] have presented interesting work related to the IBM Technologies. They used the Security Infrastructure Model (SIM) to generate the Authentication Security Policy in the WS-Security Policy standard. They used template to generate the executable security policies for IBM-WAS (IBM WebSphare Application Server). They introduced an intermediate transformation model, so that a security policy can be transformed into a variety of configuration, including IBM-WAS Deployment Descriptor. Y. Nakamura et al. [12] defined a few security objectives to be modelled in the UML language; they also described the transformation rules to transform it for some specific platform. They are focusing IBM Kerberos. Their work is more technology centric, especially the IBM technology and has only a little to do with security extensions in business process modelling and refining the PIM to further steps of the Model Driven approach.

The whole discussion regarding security modelling in early phases of software development is summarized in Table 2.5.

Table: 2.5 Related Work Regarding the Security Modelling During Early Phases of Software Development (Continue)

S/ No	**Researchers**	**Phenomenon Focused in related Work**					
		Security Objectives Focused	**Proposed Work**	**Target Architecture**	**Focused Modelling Language**	**Focused System Aspect**	**Focused Professional**
1	David Basin et al. [32, 93]	Role Based Access Control	Security Policies SecureUML (UML profile)	Not Mentioned	UML	Static System Aspect e.g. UML Class diagram	Business Domain Expert having Strong Security Knowledge
2	Jan Jürjens et al. [134, 135]	Covering Different Mechanism for Distributed Systems	UML-Sec (UML Profile)	Distributed Systems	UML	Whole System Development Process	Developer/Security Expert
3	Michal Hafner and Ruth Breu [28, 38]	Role Based Access Control	SECTET-DSL	SOA	UML	Work-Flow Security (UML Activity Diagram)	None
4	M. Memon [139]	Authentication, Non-repudiation	Security Pattern	SOA	UML	Work-Flow Security (UML Activity Diagram)	Business Domain Expert having Strong Security Knowledge
5	Alfonso Rodríguez et al. [4, 56]	Non-repudiation, Attack harm detection, Integrity, Privacy, Access control	Domain Specific Language	Not Mentioned	UML and BPMN	Business Process Modelling (UML Activity diagram and BMPN)	Business Department
6	Christian Wolter et al. [8, 30]	Confidentiality, Integrity, Authentication and Authorization, Availability and Auditing	Security Policy, Policy Constraint Model	SOA	No Specific Language	Business Process Model	Business Domain Expert

Table: 2.5 Related Work Regarding the Security Modelling During Early Phases of Software Development

S/ No	Researchers	Phenomenon Focused in related Work					
		Security Objectives Focused	**Proposed Work**	**Target Architecture**	**Focused Modelling Language**	**Focused System Aspect**	**Focused Professional**
7	Michal Menzel et al. [20]	Authentication, Authorization, Trust, Data Integrity and Data Confidentiality, System Integrity and System Availability	Security Policies.	SOA	BPMN	Business Process Model	None
8	Michal Menzel et al. [36]	User Authentication, Non-repudiation, Identity Provisioning, Data Authenticity, Data Confidentiality, Trust	A DSL Called SecureSOA, They Use the Policy Metamodel	SOA	Fundamental Modelling Concepts (FMC), Compositional Structure Diagrams	Business Process Model	None
9	Siming Kou et al. [6]	Security Requirement about different system's aspects: Stored Information, Information in Transmission, Single Service, Composite Service, Organization	They have extended SoaML for security	SOA	UML	Whole SOA System	None
10	Bertino et al. [141]	Role Based Access Control (Authorization)	They have extended WS-BPEL	SOA	BPEL	Workflow Technology	Developer /Security Expert
11	F. Satoh et al. [39, 53]	Authentication (Single Sign on, ID propagation). Security Pattern (Access Control)	MD Framework for Authentication	SOA	UML	Whole SOA system	Developer
12	Y. Nakamura et al. [12]	Authentication, Non-repudiation, Integrity, Confidentiality	Tooling framework for IBM WAS(WebSphare Application Server)	SOA	UML	Security Intention at Operation Level	Developer

2.11.2 Web Services Composition Frameworks

In this section, related work is presented about the different methods/models/frameworks presented by different researchers for web services composition. There are several terms used in literature for web services composition e.g. web-services orchestration and choreography, business process modelling or workflow modelling etc. [24].

Bart Orriens et al. [50] have presented a phased approach for services composition and named it the "*Services Composition Life cycle*". Four broad phases are described for services composition namely the definition, scheduling, construction and execution. In this approach, the UML is used for modelling the services composition; it will enable the development of technology independent composition definitions, which can subsequently be mapped to a specific services composition standard e.g. the BPEL. It is a general framework just broadly describing the process of services composition.

Roy Grønmo and Ida Solheim [23] have described the whole process of web services composition by naming it as "*Actions to build a composite web service*". The four actions they have identified for the web services composition are: discover Web services, model a composite Web service, implement the composite Web service and publish the composite Web service. They emphasized, for the services composition modelling, one should perform two kinds of modelling; service modelling and workflow modelling. Service modelling identifies services to be exposed with their interfaces and operations (UML class diagram); while, the workflow modelling identifies the control and data flows from one service to the next service (UML activity diagram). The focus of their work is workflow modelling of the composite Web service using the UML Activity diagram. Proposed framework is close to this work; however, the organization of steps/phases is done in more suitable way in the proposed framework. In the proposed framework, working is in the same direction, i.e. for service modelling, the UML class diagram is used and for workflow modelling the UML activity diagram is used.

The "UML-S" (UML for Service) is presented by Christophe Dumez et al. [21], which are basically transformation rules from UML to BPEL. They defined the static aspects of the composition i.e. the interface of the services composition by the UML-

Class diagram (WSDL (Web Services Description Language) interface and data types involved), and used the UML-activity diagram to model the dynamic aspects (the composition scenario itself, i.e. the interaction among the existing services). Christophe Dumez et al. [85] presented the different steps under the titled of "*Composite Web Service Development Process*" which should be performed for the Web services composition. Christophe Dumez in his PhD dissertation [51] presented a framework for services composition based on these steps. In the proposed framework, working is along the same direction, i.e. for services composition modelling a UML class diagram is used and for composition scenario modelling a UML activity diagram is used, however the organization of steps/phases is different.

David Skogan et al. [24] have presented an approach where services composition is modelled using a UML activity diagram. They proposed "*a method, a UML profile and transformation rules*" that can be used to produce UML models of Web services compositions. They have provided a way to model the coordination and the sequencing of the interactions among Web services. However, in this approach, methods, input/output and data transformation are modelled as notes (i.e. comments) on the side of the workflow, which can get quite confusing when the composition flow gets complex.

All of the above Frameworks do not treat security as a separate activity; the following are a few frameworks which also include security.

Jun Han et al. [3] have presented a framework named the "*Framework for security-oriented system composition and evolution*". In this framework, they have defined security at two different levels i.e. 1) System-level; which defines the security requirements of the overall system and; 2) Service-level; which defines the security requirements for a particular service. They did not discuss anything about the business process modelling and which modelling language would be used, what essential security objectives of the SOA environments are to be modelled, how these security objectives would be incorporated in the business process model, or how these security objectives would be transformed into implementations. Their focus is service security and they just provide general guidelines for secure services composition having no discussion concerning the technologies and standards used to achieve them.

Andre R. R. et al. [47] have incorporated security along the services composition and presented a methodology called the "Sec-MoSC" (Security for Model-oriented Services Composition). In this methodology a total of thirteen steps are performed in three different levels, namely the Business-level, Design-level and Execution-level. Security requirements are represented in different views corresponding to these levels. A business process model is enriched with security by adding three thing; NF-Attributes, NF-Statements and NF-Actions. They have identified security requirements and presented general guidelines for the corresponding implementation methods. They have used the BPMN as a modelling language and the BPEL for services composition. As a business process expert is not a security expert, it cannot be expect from him to incorporate too many security details. Furthermore, the beauty of a model is its simplicity, if too many details e.g. NF-Attributes, NF-Statements and NF-Actions; are added for just one non-functional attribute "security", then the whole model will become unreadable.

The whole discussion regarding the Web services composition frameworks is summarized in Table 2.6.

Table: 2.6 Related Work Regarding the Web Services Composition Frameworks

S/No	Researchers	Characteristics of the MDS Web Services Composition Frameworks				
		Title	**Phases**	**Modelling Language**	**Focused System Aspect**	**Incorporated Security Modelling**
1	Bart Orri¨ens et al. [50]	Service Composition Life Cycle	Four phases naming: Definition, Scheduling, Construction and Execution	UML	Business Process	No
2	Roy Grønmo and Ida Solheim [23]	Actions to build Composite Web Services	Four Phases naming: Discover Web Services, Model Composite Web Service, Implement Composite Web Service in a Workflow Engine, Publish Composite Web Service	UML Class Diagram, UML Activity Diagram	Business Process	No
3	Christophe Dumez et al. [85]	Composite Web Service Development Process	Five Phases naming: Import WSDLs of Web Services, UML Class Diagram generation, Generation of the interface for composite Web Service, Composite Web Service Method Definition, Composite Web Service Code Generation	UML Class Diagram, Activity Diagram	Business Process, Business-to-Business (B2B)	No
4	David Skogan et al. [24]	A Method	Four Steps Naming: UML Modelling and Searching of Web Services in the repository, Transformation of WSDL into UML Models for composite Web Service, Transformation of UML Models into XML document and into some execution engine, Publish the Composite Web service	UML Activity Diagram	Business Process, Modelling, Work-Flow Modelling	No
5	Jun Han et al. [3]	A Framework for Security Oriented System Composition and Evaluation	They defined three stages for service composition & evaluation naming: System Architecture Design, System Instantiation and Composition, and System Execution and Evaluation	No	No	Define Security at System Level and Service Level
6	Andre R. R. et al. [47]	A Methodology named Sec-MoSC (Security for Model-oriented Service Composition)	They defined thirteen steps for service composition development and grouped them in three abstraction levels, Business, Design, and Execution	BPMN	Business Process Modelling	Incorporate Security at Business process model which will propagate during the remaining steps

2.12 Tools and Technologies used

Following is a brief description of the tools and technologies used during this research process.

2.12.1 Unified Modeling Language

The UML [142] has been used during this research work for business process modelling and security modelling. Business process modelling is performed using UML activity diagram and security modeling is performed using UML-SOA-Sec. The UML-SOA-Sec is defined using UML profiling extension mechanism. For this purpose a UML class diagram is modelled to represents the metamodel of the UML-SOA-Sec in which stereotypes are defined for security objectives.

2.12.2 MagicDraw UML Modeling tool

The MagicDraw [143] UML modelling tool has been used for the UML modelling during this research work. MagicDraw is a meta CASE tool which support the definition and usage of stereotype.

2.12.3 Netbeans BPEL Designer

The Netbeans environment provides the BPEL designer. The BPEL engine is used for designing, executing and deploying BPEL workflows [144]. The BPEL is a programming-language like the XML-based declarative language, which provides basic control structures. The BPEL relies on Web service interface i.e. WSDL description of Web services [139], and acts as a container for web services.

In this work, multiple Web services are integrated in the Netbeans BPEL editor as a workflow model and this BPEL workflow is validated by the XML checker present in the same editor.

2.12.4 Java Business Integration (JBI) Standard

The JBI [145] is a standard which provides various pluggable components, which can be deployed as JAR files. The JBI provides a platform for integrating applications using WSDL and XML-based messaging. The JBI uses WSDL 2.0 as a service/message model. The BPEL workflows are integrated in the JBI environment.

The top level architecture of the JBI is shown in the Figure 2.11 [145]. As can be seen, two types of components are presents in the JBI: Binding Components (BCs) and Service Engines (SEs). External business partners are plugged into the JBI containers through the BCs. For Enterprise Application Integration (EAI), the service consumer and service providers are connected to the Enterprise Service Bus (ESB) through the BCs such as the SOAP, JMS, and HTTP etc. The SEs provides integration of application and performs transformation and routing. The SEs such as WS-BPEL, Apache Camel etc. are basically business logic drivers and can act as service consumers or service providers or both. The JBI deals with the installation, deployment, monitoring and control of the BCs and SEs and defines packaging standards for them.

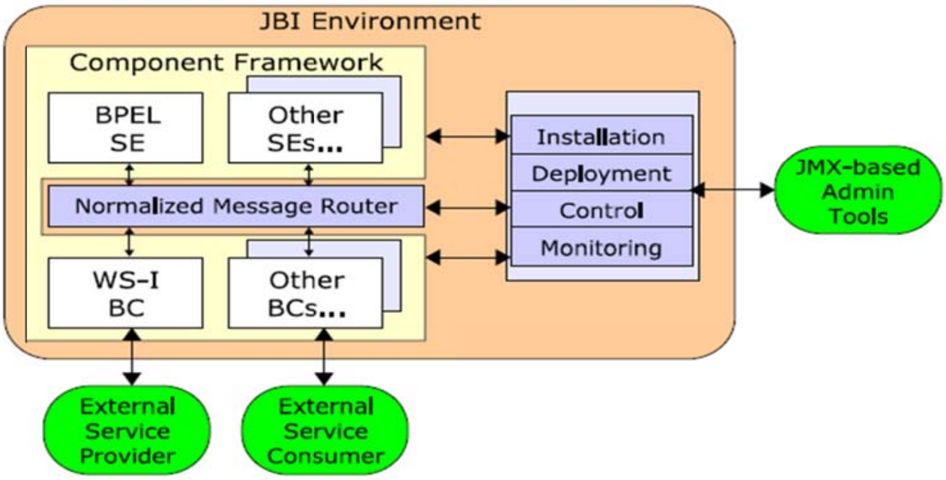


Figure 2.11: Top-level View of the JBI Architecture

The Normalized Message Router is another important component of the JBI. It is the central delivery mechanism which provides the loosely-coupled message exchanges between the SEs and BCs deployed with in a JBI Runtime.

In this research work, the BPEL workflow is integrated in the JBI environment.

2.12.5 Apachi Tomcat Server

The Apachi Tomcat Server is an open source application server from the Apachi Software Foundation. An application server can be defined as *"a software framework that provides an environment in which applications can run, no matter what the applications are or what they do"* [146]. Basically it serves as a web application container. Apache Tomcat powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations [147].

During this research work, the Apachi Tomcat server is used to deploy the EIS composite application.

2.13 Chapter Summary

During this work the focus is security modeling along the business process modeling of SOA applications using MDSD approach. That's why; this chapter provides details about the necessary background knowledge which support the realization of this research work. It defined foundational concepts of Web services based SOA systems, services composition and their standards/ languages, and SOA Security and Business Process Modelling. It also gave an overview of the Model Driven Software Development and the Domain Specific Language. Later on, this chapter described the security objectives focused on by different researchers in their works. First, it described those security objectives where authors did not mention the target environment; afterwards, it described those security objectives where authors mentioned specifically the SOA environment. Afterwards it provides the detailed discussion about finding the essential security objectives to be modeled in a business process model of an SOA application. Later on, it presented the related work carried out by different researcher in the same area. At the end, it presented the overview of the tools and technologies used for the prototype implementation.

CHAPTER 3
RESEARCH METHODOLOGY

3.0 Chapter Overview

The purpose of the chapter is to explain the choice of following particular research methods during this research work. Initially, a brief account about the research methods used in the field of software engineering is presented. After that, research activities performed during this research work are described in detail. Later on, a description is presented regarding the research methods used during this research work followed by the description of techniques used for research data analysis.

3.1 Research Methods in Software Engineering

The field of software engineering involves development, operation and maintenance of software. The aim of the software engineering research is to investigate how software development, operation and maintenance are conducted by software engineers and other stakeholders. Research in software engineering intends to improve the practice of software development [148]. All activities of software development are conducted by human, either individually or in the form of group or organization; hence, social and political questions are important for this development [149]. Therefore, research in software engineering is always complex and difficult due to the awkward intersection of machines and humans; that's why researchers have started to study the technical and non-technical issues in software engineering [150]. Software engineering activities are not only based on the tools and processes, but it also depends on the social and cognitive processes around it [151]. Therefore in software engineering domain, study of human activities is important to understand a problem. The importance of human activities in software engineering field requires using the research method which is related to the study of human behavior. Research method of sociology become more relevant when the problem under consideration is related with person, teams and organization [151].

The software engineering research community has a practical and result oriented view on research method, rather than a philosophical stand [150]. In the field of software engineering, there is a lack of guidance regarding the selection of a particular approach to answer a particular research question [148]. Normally researchers choose qualitative or quantitative research methods for conducting research. These research methods are distinguished on the basis of the nature of data and the process followed to collect the data during the research [149, 152]. However, in software engineering due to the blend of technical and human aspects, qualitative and quantitative methods are combined, in order to take advantage of the strength of both [150]; therefore, mixed method is emerging as a third choice for software engineering researchers [153]. Description of these three research methods is provided in the following sections along with the description of the specific methods used during this research work i.e. case study and survey methods.

3.1.1 Qualitative Method

The qualitative methods are used to collect and analyze the qualitative data. Qualitative data is normally in the form of pictures, words, statements, description and diagrams. The process followed to collect them are ethnographies, case studies and interviews [152]. In qualitative methods the focus is more towards the collecting and analyzing the non-numeric data, and information are explored in depth rather than in breath [154]. It attempts to get an in-depth opinion from participants [155]. Qualitative data is analyzed using categorization and sorting [149]. Qualitative research explores attitudes, behavior and experiences and the research methodologies used are: Phenomenology, Ethnography, Case studies, Interviews, Action Research, Grounded Theory [155, 156]. Brief description of these research methodologies are discussed below:

- *Phenomenology:* This methodology involved the research discipline from social sciences such as psychology, sociology and social work. Its focus is experience of different people's and their interpretation about the world. This approach is based in a paradigm of personal knowledge and subjectivity, and emphasizes the importance of personal perspective and interpretation.

- *Ethnography*: Ethnography is the study of people and their cultures through close observation. It is basically originated from the anthropology. This method involved an extensive field work in the culture which is under study. Ethnography research method is used in many disciplines like political and social studies, anthropology and education.

- *Case Studies:* Case study is " *an empirical inquiry that investigate a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clear*" [157]. Case studies are used to generate the detailed insight of the particular case, its processes and relationship [157]. Case studies can be used in both qualitative and quantitative research methods [158].

- *Action Research*: The aim of the action research is "*learning by doing*". A group of people identify a problem, do something to resolve it, see how successful their efforts were, and if not satisfied, try again. There are several names can be used for action research such as participatory research, collaborative inquiry, emancipatory research, action learning [159]. Action research is used to contribute and improve researcher's strategies, practices and knowledge of the specific case about which he/she is practicing [160]

- *Grounded Theory*: Grounded theory is defined as "*to generate or discover a theory*" [161]. It can also be defined as "*the discovery of theory from data systematically obtained from social research*" [161]. In grounded theory the intension is to do the field research and collect the data and then analyse the data and find out what theories can be emerge so that theory is grounded in the field data. Theories that are captured from the data are helpful to resemble what is going on.

In qualitative research community, validity is used to express the quality differences of research studies [162]. It is claimed that the qualitative methods cannot be generalized because of having detail information on human, social and cultural phenomena [163]. On the other hand, results of the qualitative studies cannot be considered invalid on these grounds [164]. In qualitative research, six types of validity are considered [162]: descriptive, interpretive, concurrency, internal, external and theoretical validity.

3.1.2 Quantitative Method

Quantitative method is a formal, objective, systematic process in which numerical data are used to obtain information about the world [165]. The quantitative methods are used to collect and analyse the data which is in the form of numbers. Quantitative methods are generally designed to collect data in a form suitable for statistical analysis and data should be collected through standards measures [166]. Alan Bryman [167] described the five ways for collecting data naming: social survey, experiments, analysis of previously collected data, structured observation, and contents analysis. Later on, statistical analysis may be applied on quantitative data to interpret the results with the help of different charts or diagrams. Quantitative research generates statistics through the use of survey research, using methods such as questionnaires or structured interviews [152, 155]. There are several quantitative research designs techniques [165], their brief description is presented below:

- Descriptive research: It describes what exists and may help to uncover new facts and meaning. The purpose of descriptive research is to observe, describe and document aspects of a situation as it naturally occurs [168].

- Correlational research: The aim of the quantitative correlational research is to systematically investigate and explain the nature of the relationship between variables in the real world.

- Experimental research: In experimental research design cases and effect relationship is studied.

- Quasi-experimental research: Quasi-experimental research is similar to the experimental research; however, it does involve the manipulation of an independent variable.

- Survey research: A survey is used to obtain information from groups of people (i.e. populations) [168]. Data can be collected using the tool such as interviews or questionnaires. The information that is obtained may be concerned with the prevalence, the distribution, and/or the interrelationships between variables within these groups.

- Evaluation research: This research is an "*applied form of research that involved finding out how well a programme, practice, procedure or policy is working*" [168].

As quantitative data is in the form of numbers that's why quantitative results are based on statistics. The validity of the statistical results are referred to the validity of relationship of two variables and strength of the relationship of these variables [162].

3.1.3 Mixed Method

In the field of software engineering due to the blend of technical and human aspects, qualitative and quantitative methods are combined in a single study, in order to take advantage of the strength of both [150]. The combination of quantitative and qualitative approaches in a single study is referred as mixed method studies which is emerging as a third choice for software engineering researchers [150, 153, 169]. In the mixed method, two different types of data (*data in the form of numbers and the data in the form of text or pictures*) are integrated at several research stages e.g. data collection, data analysis, data interpretation [156]. Using the combination of both methods is beneficial that it provides information from number of prospective and it provides possibly more generalized and reliable result [170].

Green et al. [171] have highlighted five purposes of mixed method studies:

1. Being Triangulation: convergence of results, rationale for triangulation is to increase the validity of results by using different methods [172]

2. Complementarity: "*Seeks elaboration, enhancement, illustration, clarification of the result from one method with the result from the other method*" [171].

3. Development: which uses the result of one method to develop or inform other method.

4. Initiation: It involves recasting of questions or results from one method with the questions or results from the other method.

5. Expansion: Different inquiry components are inquired by different methods

As mixed method studies make use of both methods i.e. quantitative and qualitative in a single study; therefore, all types of qualitative and quantitative validity are relevant to the mixed method studies [162].

3.1.4 Case Study Method

In the field of software engineering, case study is a suitable research method since it studies contemporary phenomena in its natural context and it provide deeper understanding of the phenomena under study [149]. Case studies are used to generate the detailed insight of the particular case, its processes and relationship [157]. In the field of software engineering, case study method is used for the evaluation of software engineering methods and tools before it has to be used on a "*real*" software project [173]. It is studied that, case studies have been carried out in software engineering domain for validating the outcomes of the research studies in which researchers validate their work by applying it on the real world problems [28, 139, 174].

3.1.5 Survey Method

Software engineering activities are not only based on the tools and processes, but it also depends on the social and cognitive processes around it [151]. Therefore in software engineering domain, study of human activities is important to understand a problem. The importance of human activities in software engineering field requires using the research method which is related to the study of human behavior; hence, "*survey method*" is a potential method for software engineering research. Survey can be defined as "*a comprehensive system for collecting data using a standardized questionnaire*" [175]. The data collected through survey are used to "*describe, compare or explain knowledge, attitude and behavior*" [175].

In software engineering research, survey method can also be used for the evaluation purposes. During the survey process, respondents, who have the knowledge/experience of the specific methods or tools, are asked to provide information about the method or tool. These information from the respondents are analysed using standards statistical techniques There are several advantages of using the survey method for evaluation purposes e.g. it makes use of existing experience, it can confirm that an effect generalises to many projects/organisations and it uses of standard statistical analysis techniques [176].

Kasunic [175] has explained a comprehensive seven step process for conducting a survey:

- Identification of aim
- Identification of target audience
- Design of sampling plan
- Questionnaire formulation
- Pilot test of questionnaire
- Questionnaire distribution
- Analysis of the results

3.2 Present Research Activities

Research methodologies are basically a framework of overall research activities. The overall research work is divided into four phases naming: Development of proposed DSL"UML-SOA-Sec", development of proposed "Saleem's MDS services composition framework", validation of the proposed work and comparative study and evaluation of the proposed work. Each phase is concerned with the specific goals to finally fulfill the main objectives. Each of these four phases has different activities which are pictorially represented in Figure 3.1 and described in the following subsections:

3.2.1 Development of Proposed DSL "UML-SOA-Sec"

The research started with the literature study of the security issues faced by SOA applications. It has been found that the Model-driven Security is an interesting approach for the design and development of SOA applications. However, there is a lack of modelling techniques for modelling security objectives during the business process modelling of SOA applications. In this connection a DSL, "UML-SOA-Sec" is developed for modelling the security objectives during the business process modeling of SOA applications.

1. For the identification of essential security objectives to be modelled for SOA applications, a thorough literature study is performed. Work of the different authors is analyzed where they modelled the security objectives along the business process modelling. Firstly, a general literature study is performed, where authors did not mention the architecture of the application weather it is an SOA or not. Afterwards, a literature study of those applications is performed where authors specifically mentioned the architectural style as an SOA. The focus is to find out different security objectives necessary to be modelled for SOA applications. Afterwards, a critical analysis of these security objectives is performed for their suitability to be modelled in a business process model for SOA applications and selection of the most appropriate security objectives.

2. A DSL "UML-SOA-Sec" is developed based on the security objectives identified in the previous step. An abstract syntax of the DSL is defined by a metamodel and a concrete syntax is defined by a UML-profile. The UML modelling language is used during this research work, which is an industry standard for business process modelling [4]. The UML-profiling mechanism is used to extend the UML to incorporate security objectives, which allows the specification of security objectives while business process modelling [126]. The detail discussion about the development of "UML-SOA-Sec" is described in section 4.1 of chapter four.

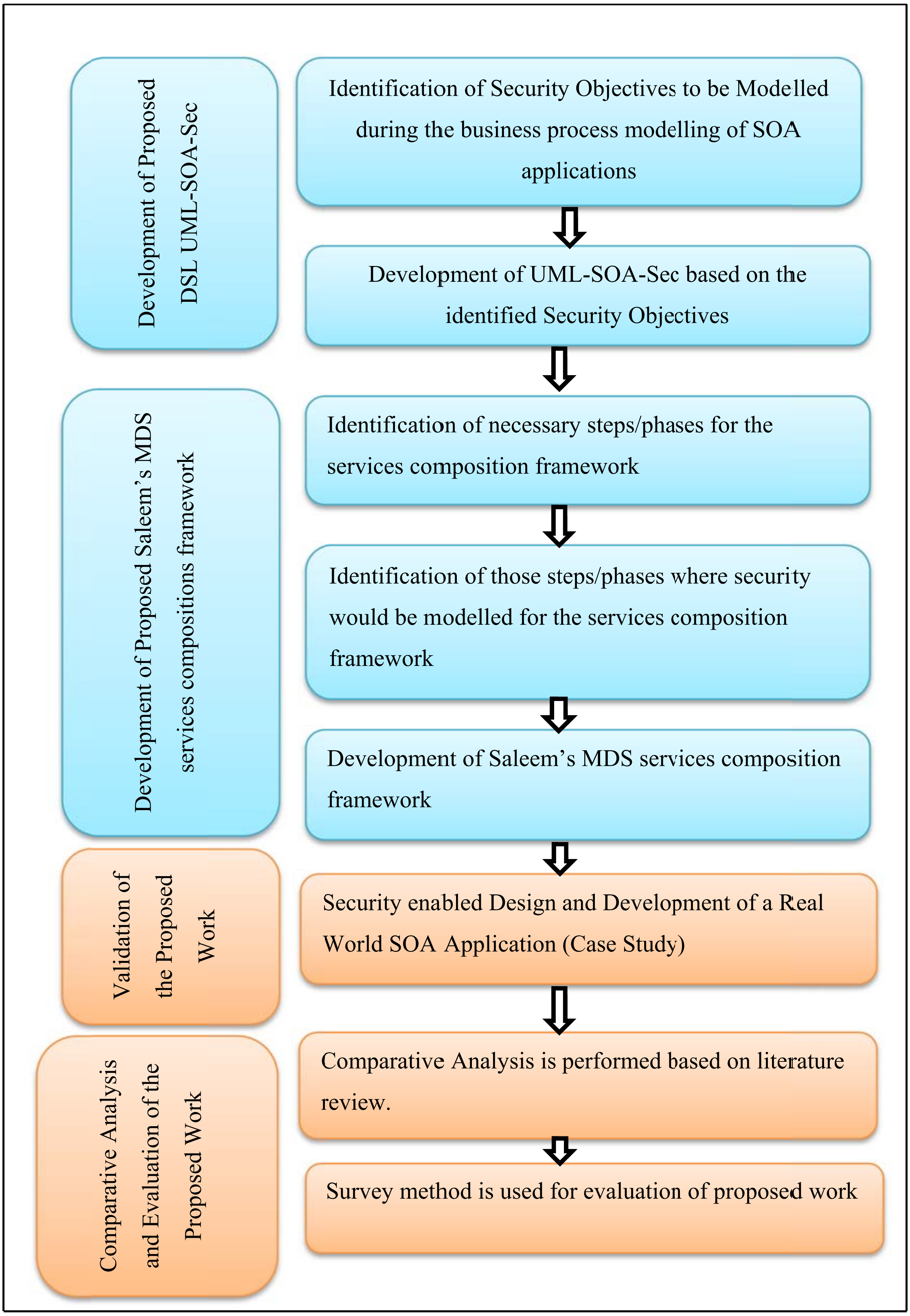

Development of Proposed DSL UML-SOA-Sec
Identification of Security Objectives to be Modelled during the business process modelling of SOA applications
Development of UML-SOA-Sec based on the identified Security Objectives
Development of Proposed Saleem's MDS services compositions framework
Identification of necessary steps/phases for the services composition framework
Identification of those steps/phases where security would be modelled for the services composition framework
Development of Saleem's MDS services composition framework
Validation of the Proposed Work
Security enabled Design and Development of a Real World SOA Application (Case Study)
Comparative Analysis and Evaluation of the Proposed Work
Comparative Analysis is performed based on literature review.
Survey method is used for evaluation of proposed work

Figure 3.1 Present Research Activities

3.2.2 Development of Proposed Saleem's MDS Services Composition Framework

SOA applications are actually a composition of services. It has been found that there are many frameworks proposed for the Model-driven development of services compositions containing several steps/phases (ranging from four to thirteen). However, there are no clear identifications of the most necessary steps/phases for service composition framework. Furthermore, notion of security is neglected in almost all of these frameworks i.e. security is not defined during the business process modelling of SOA applications developed through these frameworks. In this connection, an MDS service composition framework is presented for secure web services composition.

- For the identification of necessary steps/phases for the services composition framework, a thorough literature study is performed. Several services composition frameworks/models/methods are studied and four essential steps are identified which should be included in a services composition framework.

- Later on, those steps of the MDS services composition framework are identified where security would be modelled along the business process modelling for a secure services composition.

- Saleem's MDS services composition framework is developed for the secure web services composition based on the steps identified in the previous section. The detailed discussion about the development of "*Saleem's MDS services composition framework*" is described in section 4.2 of chapter four.

3.2.3 Validation the of Proposed Work

In the context of the current research study, case study method is used for the validation of the proposed work which helped in studying proposed phenomenon in its natural context and provided deeper understanding of it. This research work has two contributions i.e. proposed DSL "*UML-SOA-Sec*" and "*Saleem's MDS services composition framework*". The first contribution i.e. proposed DSL "UML-SOA-Sec" is validated by applying it on a business process model of the SOA application. The security annotated business process model is created using UML Activity diagram and annotated with security objectives defined in the "UML-SOA-Sec". The second contribution i.e. "Saleem's MDS services composition framework" is validated with the prototype implementation of the SOA application. A secure composite web service is developed and deployed using the open source tools and technologies. Detailed discussions are provided in section 3.3.1 of this chapter.

3.2.4 Comparative Analysis and Evaluation of the of Proposed Work

- Comparative analysis of proposed work is performed with the related work. Firstly, comparative analyses are presented about the "UML-SOA-Sec" and the research works very close to it. Secondly, comparative analyses are presented about the "Saleem's MDS services composition framework" and other MDS services composition frameworks. Findings are represented in the form of discussion as well as tables. The detailed discussion about the comparative analysis is provided in section 7.1 of chapter seven.

- Later on, to evaluate the proposed work, survey method is used. The objective of the survey method is to evaluate the proposed DSL "UML-SOA-Sec" with the previously proposed DSLs [4, 8, 28, 30, 36, 38, 56, 139] on the basis of approaches used in these DSLs for security annotation to find out the respondents response upon improvement in annotating the business process diagram with security using "UML-SOA-Sec". Personally administered questionnaire are used as a survey instrument for data collection. Data is analyzed and findings are presented in the form of discussion as well in the form graphs. Detailed discussions about the survey conducted during this work are provided in section 3.3.2 of this chapter.

3.3 Validation Methods for the Overall Study

Mixed method is used as a main method during this research study in which qualitative and quantitative methods are combined in a single study. In mixed method, a problem is approaches from different perspectives that why it provides additional insight [177] and it prevent from biasedness problems either related to the methods or researcher data sources [178]. The purpose of mixed method is "*to obtain different but complementary data on the same topic*" [179]. The multiple type of data sources make finding that could not be made by using a single data source and it also increase the confidence in finding [180]. In software engineering field due to involvement of technical and human aspect, qualitative and quantitative methods are combined in order to take strength and advantages of both [150]. The most important feature of the mixed method which increases its suitability for the current research work is that it is useful for attempting to confirm and cross-validate study findings [181]. Therefore, based on the advantages of the mixed method approach, justified by the literature, it is chosen as a research method in the present research study.

During this research study, a case study method (qualitative) is used as the primary method to validate the proposed work i.e. proposed DSL "UML-SOA-Sec" and proposed framework "*Saleem's MDS services composition framework*". Afterwards, a survey method (quantitative) is used to strengthen the results of the case study and to evaluate the proposed work.

There are two approaches for data collection during the execution of qualitative and quantitative methods named as *sequential* and *concurrent*. In sequential approach one method is given priority over the other method and executed first while in concurrent approach both methods are given equal status and executed simultaneously [181]. According to the requirement of the present research work sequential approach is adopted where qualitative method (case study) is performed first and afterwards quantitative method (survey) is performed. The detail discussion about these two methods is described in the following sections.

3.3.1 Case Study: Security Enabled Design and Development of an SOA Application.

For the validation of proposed work i.e. "UML-SOA-Sec" and "*Saleem's MDS services composition framework*", a real system of SOA environment is required. In an SOA environment, a system is composed of services offered by different partners. These services are deployed on different servers which are located at different sites. Each site contains its own database and performs both roles, i.e. service provider as well as service consumer. These services from different partner organizations are integrated/composed to form application and every organization work independently without any central control [182]. The distributed nature of the SOA environment and technology-heterogeneity of the services, raise many security challenges. The inter-organizational workflow in the SOA environment is executed in a decentralized manner where one need to secure local data stores as well as the communication among the partners [139]. In such environments different use-cases can be derived from the scenarios to model the business process of an SOA-based system [183, 184].

The proposed DSL and proposed MDS services composition framework are general and cater for the requirements of different service-oriented domains like e-government, e-health, e-education etc. For demonstration of this research work, the SOA based healthcare scenario is used which is a Healthcare Service Specification Project (HSSP) specified service called the Entity Identification Service (EIS) [185]. The detailed discussion about this particular case is provided in chapter five. Hence, a case study of the real world SOA application is selected for the purpose of validation. It is studied that, case studies have been carried out in software engineering domain for validating the outcomes of the studies [28, 139, 174]. In the context of the current research study, case study method helped in studying proposed phenomenon in its natural context and provided deeper understanding of it. Hence, before proposed work will be used on a real software project, the case study method is used for the validation of the proposed work.

This research work has two contributions i.e. modeling of security objectives in a business process model for SOA applications and secure web services composition. The first contribution i.e. proposed DSL "UML-SOA-Sec" is validated by applying it on a business process model of the case study. The security annotated business process model of the case study is created using UML Activity diagram and annotated with security objectives defined in the "UML-SOA-Sec". The second contribution i.e. "Saleem's MDS services composition framework" is validated with its prototype implementation. A secure composite web service is developed and deployed using the open source tools and technologies. The whole process of secure business process modeling and composite application development is described in chapter five and six respectively.

3.3.2 Survey: Evaluation of "UML-SOA-Sec"

Following subsections illustrate the detailed descriptions of the survey method used for the evaluation of the "UML-SOA-Sec".

3.3.2.1 Objective of the Survey Method:

The objective of the survey method is to evaluate the proposed DSL "UML-SOA-Sec" with the previously proposed DSLs [4, 8, 28, 30, 36, 38, 56, 139] on the basis of approaches used in these DSLs for security annotation to find out the respondents response upon improvement in annotating the business process diagram with security using "UML-SOA-Sec".

3.3.2.2 Evaluation factors for DSLs

Different DSLs proposed by different researchers adopted different approaches to annotate the business process model with security. Some are proposing icons [30, 186], while on the other hand, some are just proposing security stereotypes (textual description) [28, 29, 55]. Furthermore, few are proposing multiple diagrams to represents the business process model [8, 28]. Moreover, the number of security objectives present in different security DSLs, are also different. The above mentioned factors affect a business process model when it is annotated with security by following a specific technique presented in the DSLs. About these factors we have collected the data and performed analysis.

3.3.2.3 Evaluation criteria for DSLs

The evaluation was carried out on the basis of the dimensions of the success criteria defined by Roy Gronmo [23]. The dimensions of the success criteria which are measured through the questionnaire are: simplicity of business process model, readability of business process model, and ease of use by business process expert, sufficient number of security objectives present in the DSL for SOA environment.

3.3.2.4 Experiment Design: Evaluation of the Security Annotating Techniques of DSLs in a Business Process Model

In the evaluation process, initially, a qualitative experiment is performed in which feature-based evaluation is performed by a group of potential user who are expected to try out the methods on business process model of the case-study before making their evaluations [176]. In the current research work, business process model of the case-study was modelled using the UML Activity diagram. Respondents are requested to add the security in this business process model using the UML-SOA-Sec. Side by side respondents are presented with techniques of the other researchers for annotating the business process model with security.

Afterwards, a quantitative survey [176] is performed for the evaluation purposes in which when the respondents finish with the process of adding security in the business process model then they were presented with the questionnaire in which they have to record their feedback against different techniques adopted by these researchers in annotating the business process model with security. Initially the feedback of the respondents is organized in tables according to particular researcher. Later on for evaluation purpose the data is organized in tables according to particular questions. Figure 3.2 describes the whole process.

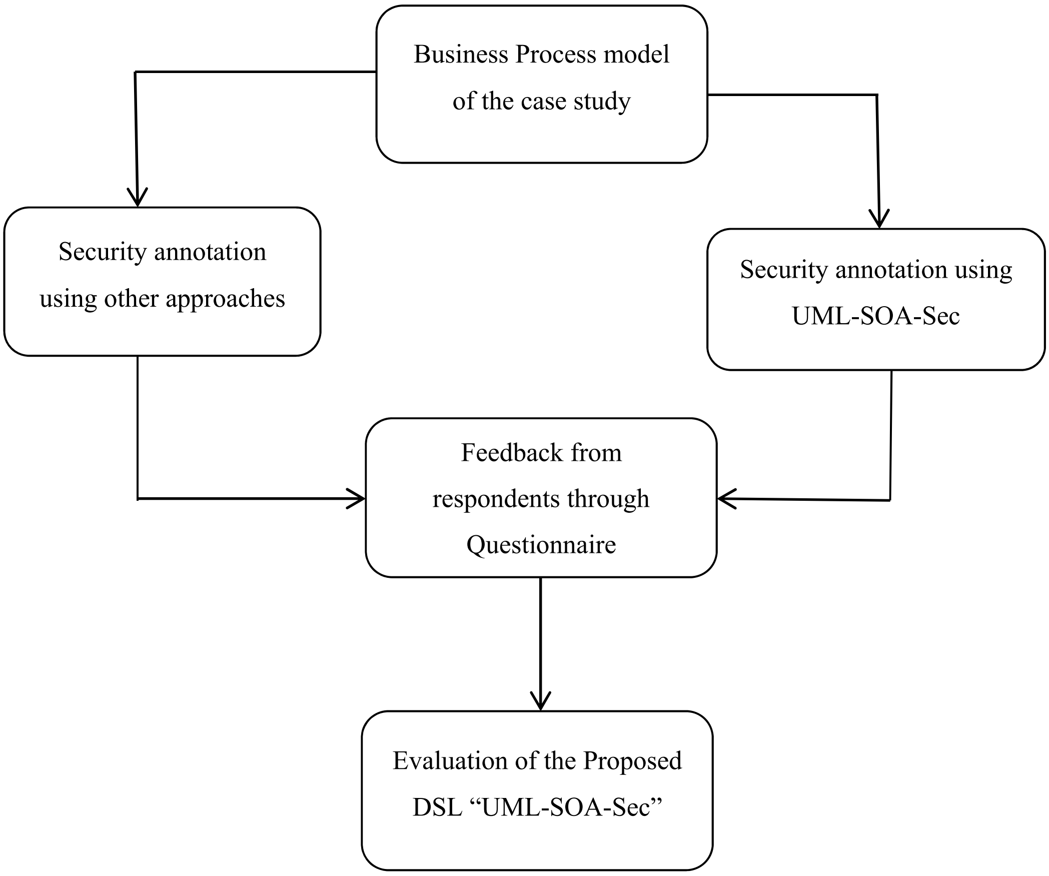

Business Process model of the case study
Security annotation using other approaches
Security annotation using UML-SOA-Sec
Feedback from respondents through Questionnaire
Evaluation of the Proposed DSL "UML-SOA-Sec"

Figure 3.2 Evaluation Procedure

3.3.2.5 Instrument

The survey instrument used in this study is personally administered questionnaire. In the context of this study, due to time and cost constraints, it was necessary to find a survey instrument which consumes less time and cost. That's why personally administered questionnaire is used as a survey instrument [187]. The reasons for not adopting other data gathering techniques such as interviews and observations are time, cost, non-availability and disinterest of respondents and confidentiality [187].

3.3.2.6 Scaling

During this work, ordinal scale is used to collect the response from the respondents. The ordinal scale is capable of describing the order. In ordinal measurement, numbers are assigned to the objects and these numbers represents the rank or order of the category. The order of the number is of the interest in ordinal scale instead of the number itself [188]. Five point Likert scale, presented by Rensis Likert [189] is one of the example of ordinal scale [175].

The perfect number of points in a Likert scale has not achieved consensus among researchers. Studies have shown that respondents feel inconvenient to respond to a Likert scale of more than seven points [190], so any number lesser then seven is suitable. The two most prominent reasons to have a five point Likert scale are [191]: Firstly, having a neutral feeling about a statement or a topic is natural and legitimate among respondents. Not providing a neutral point to respondents can force them to answer positively or negatively, which results in biased answers. Secondly, the mid-point in five point Likert scale, which is 3, is *"right in the middle"* and perfectly denotes a mixed feeling. Moreover, the originator of the scale, Rensis Likert [189], proposed a five-point Likert scale. That's why five point Likert scale is widely used in conducting the survey and also used during this research work.

In Likert scale questions are also changed into statements and respondents are asked to indicate their level of agreement accordingly [192]. Furthermore, the use of this scale makes the questionnaire simple to respond and easy to analyse the data.

During this research, questions designed are close-ended at five point Likert scales; where respondents are provided a set of answers from which they have to choose as shown in Figure 3.3.

S/No	Questions	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

Figure 3.3: Five Point Likert scale

3.3.2.7 Survey Questionnaire

During this research work, questions or statements of the questionnaire are developed according to the guideline suggested by [193]. These guidelines suggest that "double-barrelled question" should be avoided i.e. in a single question, ask two different but possibly related questions. Questions that use negative words should be avoided too because they may be the possible cause of confusion for the respondents. Furthermore, statements or questions should not be biased that may lead to biased responses. Moreover, the sequence of questions should be logical and easy to follow for the respondents.

The questions are developed on the basis of the dimensions of the success criteria defined by Roy Gronmo [23]. The five dimensions of the success criteria are simplicity of business process model, readability of business process model, ease of use by business process expert, the use of icons to represent the security objective, sufficient number of security objectives present in the DSLs.

There are several features in the DSLs proposed by different researchers, which affect the business process model and regarding them we want to collect the data and will perform analysis. Different researchers, adopted different approaches to annotate the business process with security. Some are proposing icons [30, 186], while on the other hand, some are just proposing security stereotypes (textual description) [28, 29, 55]. Furthermore, few are proposing multiple diagrams to represents the business process model [8, 28]. Moreover, the number of security objectives presents in different security DSLs, are also different. The above mentioned factors affect a business process model when it is annotated with security by following a specific technique presented in the DSLs. These are the factors regarding them questions are designed to get the feedback from the respondents. The questionnaire is attached as Appendix A.

3.3.2.8 Sampling

Keeping in view the research objectives of this study it is important to consider the proper sample size that supports in collecting the data accordingly. In this study, a non-probability sampling method known as convenience sampling is used. Convenience sampling is used when the data has to be collected from the population which is readily available and convenient and inexpensive [194].

The respondents were carefully selected considering their expertise in the subject concerned. Since during this study, it has to investigate that how the UML-SOA-Sec improves security modelling in business process modelling; hence, a smaller but focused samples are more often needed than large samples [195].

3.3.2.9 Respondents and their selection criteria

The respondent of this survey were the software developer working in software industry and postgraduate students of computer science department of University Teknologi PETRONAS, Malaysia. Therefore in this study, 30 respondents were involved which are being considered as a sufficient as per the central limit theorem [196]. The details regarding the respondents have been provided in Appendix B.

The criteria for the selection of respondents is that, they should have the knowledge of business process modeling using UML Activity diagram or BPMN and they were familiar with the basic knowledge of security objectives. This is because the modeling languages used in the DSLs presented by researchers is either UML or BPMN, which are considered as industry standard for business process modeling. Furthermore, they were briefed about the objectives of this study. They were presented the business process diagrams created according to the techniques adopted by different researchers for security annotation in their DSLs. This whole process of targeting participants assisted in collecting data properly relating to the research objectives of this study.

3.2.2.10 Distribution and Collection of Questionnaire

The questionnaire was personally administrated among the respondents which help in avoiding the biased data collection from the participants because it does not allow for sharing the dialogues or suggestions regarding the questionnaire elements. Feedback is personally collected from the respondents of the postgraduate students of the University Teknologi PETRONAS. Feedback from the respondents of the software industry is collected through email. This whole data collection procedure helps is getting proper feedback.

3.4 Research Data Analysis

Data analysis is conducted differently for qualitative and quantitative data. Following sections illustrate them in details.

3.4.1 Qualitative Data Analysis for Case Study Method

Different analytical techniques are used in case studies method such as pattern-matching, explanation-building, and time-series analysis. In pattern-matching techniques, an empirically based pattern is compared with the predicted pattern of the case study. If the patterns coincide, the results can help a case study to strengthen its internal validity. While in explanation-building method the data of the case study is analyzed by building an explanation about the case. Whereas in time-series analysis as cleared from its name a time-series analysis is conducted. This technique is similar to the time-series analysis conducted in experiments and quasi-experiments. The more precise the pattern, the more that the time-series analysis will lay a firm foundation for the conclusions of the case study [157, 197] .

During this study, an explanation building technique [157] is adopted that support in comparative analysis of existing related studies with the proposed research work. In explanation building, many different kinds of evidence, figures, statements, documents, are linked together to support a strong and relevant conclusion [149]. During this research work, figures (business process diagram) and statements are used as an evidence to support the conclusion.

In order to analyse the results, two comparative studies have been conducted in this research work by following the guidelines presented by Roy Gronmo [23]. The first comparative study has been conducted for “UML-SOA-Sec”. From through literature review, it is revealed that, the different DSLs proposed by different researchers, adopted different approaches to annotate the business process with security. Some are proposing icons [30, 186], while on the other hand, some are just proposing security stereotypes (textual description) [28, 29, 55]. Furthermore, few are proposing multiple diagrams to represents the business process model [8, 28]. Moreover, the number of security objectives presents in different security DSLs, are also different. These are the factors which affect a business process model and regarding them the data is collected and analysis is performed. In this study, proposed DSL is compared with the other DSLs by considering the numbers of factors such as: simplicity and readability of business process model, ease of use by business process expert, use of icons in a DSL to represent the security objectives and sufficient numbers of security objectives for SOA applications. The outcome of the comparative study is represented in the form of statements as well as table. Detailed discussions are provided in section 7.1.1 of the thesis.

In addition, second comparative study is undertaken for "*Saleem's MDS services composition framework*". In this study, proposed framework is compared with the other web services composition framework by considering the numbers of factors such as: phases/steps of the framework, modeling language used, focused system aspect and incorporating security modeling respectively. The outcome of the comparative study is represented in the form of statements as well as table Detailed discussions are provided in section 7.1.2 of the thesis.

These comparative studies reasonably supports in validating the outcome of this research study by obtaining the research objectives. For analyzing the textual data, the standard tool MS-Word is used [149] during this study that assist in organizing and analyzing the qualitative data properly.

3.4.2 Quantitative Data Analysis for Survey Method

Usually the two main types of statistical methods have been commonly used by researchers in analysing the quantitative data, descriptive and inferential statistical methods [198]. Descriptive statistical methods generally describe the data that is enumerated, organized and graphically represented. Whereas the inferential statistical methods generally draw or predict the conclusions based on given data of participants/population. Keeping in view the nature of this research study and its objectives; descriptive statistical method is selected and used for data analysis. Descriptive procedure is ideal for obtaining the distributional properties of numeric values, however, descriptive statistics cannot be generalised beyond the data at hand [199].

The objective of the survey method is to evaluate the proposed DSL "UML-SOA-Sec" with the previously proposed DSLs [4, 8, 28, 30, 36, 38, 56, 139] on the basis of approaches used in these DSLs for security annotation to find out the respondents response upon improvement in annotating the business process diagram with security using "UML-SOA-Sec". Keeping in view the objectives of this research study, the specific descriptive statistical method used during this study is *frequency distribution*. Frequency distribution is a way of displaying the data in an organized manner so that questions could be answered easily. A frequency distribution is simply a table that displays how many times in a data set each response occurs. It is useful to answer the questions with proportions or percentages. A frequency can easily be converted into percentage by simply dividing the number of times the score occurs in the data by the total number of responses [200-202].

The different DSLs, proposed by different researchers, adopted different approaches to annotate the business process with security. Some are proposing icons [30, 186], while on the other hand, some are just proposing security stereotypes (textual description) [28, 29, 55]. Furthermore, few are proposing multiple diagrams to represents the business process model [8, 28]. Moreover, the number of security objectives presents in different security DSLs, are also different. These are the factors which affect a business process model and regarding them we have collected the data and performed analysis.

In order to collect the data, a questionnaire is designed based on the success criteria defined by Roy Gronmo [23] with close-ended questions at five point Likert scales. The dimensions of the success criteria which are measured through the questionnaire are: simplicity of business process model, readability of business process model, ease of use by business process expert, use of icons in a DSL to represent the security objectives and sufficient number of security objectives present in the DSL for SOA environment.

During this research work, respondents are presented with the six models of different researchers along with the questionnaire for which they have to provide their feedback. Initially the feedback of the respondents is organized in tables according to model of a particular researcher. These tables of respondent's feedback are presented in Appendix C. Later on, for evaluation purposes, the data is reorganized according to a particular question against all the researchers. Keeping in view the objectives of this research study, the descriptive statistical method used is frequency analysis, where data collected from the survey is coded in the form of table; percentage is computed by dividing the number of times the score occurs in the data by the total number of responses. Column graphs (bar charts) are simulated using MS-Excel tool [149] to represent the data in pictorial form. It helps in describing and discussing the improvement in annotating the business process diagram with security using "UML-SOA-Sec" based on the response of the respondents [203-205]. The whole evaluation process is described in the Figure 3.4.

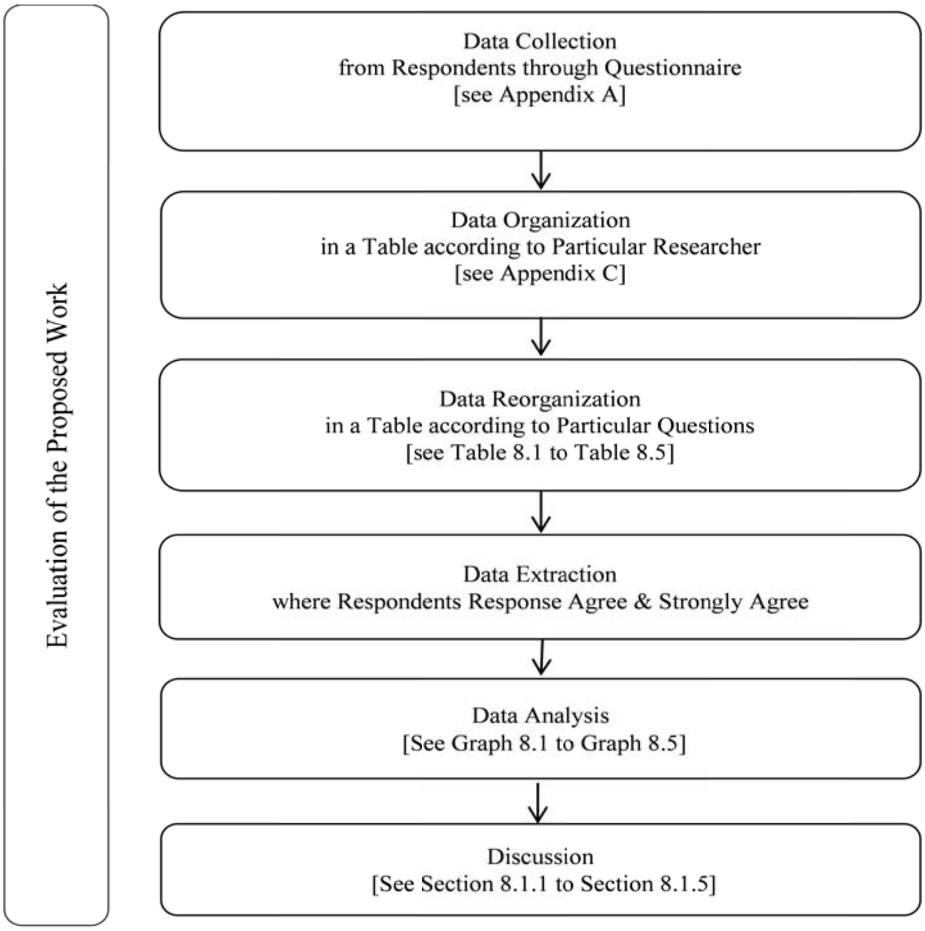

Evaluation of the Proposed Work
Data Collection
from Respondents through Questionnaire
[see Appendix A]
Data Organization
in a Table according to Particular Researcher
[see Appendix C]
Data Reorganization
in a Table according to Particular Questions
[see Table 8.1 to Table 8.5]
Data Extraction
where Respondents Response Agree & Strongly Agree
Data Analysis
[See Graph 8.1 to Graph 8.5]
Discussion
[See Section 8.1.1 to Section 8.1.5]

Figure 3.4: Evaluation Process

3.5 Chapter Summary

This chapter starts with illustrating the research in the field of software engineering and also described the different research methods used in it. After that, research activities performed during this research work are described in detail. Later on, a description is presented regarding the research methods used during this research work followed by the description of research data analysis.

CHAPTER 4
PROPOSED WORK

4.0 Chapter Overview

This chapter starts with the description of an overall view of the whole process of language definition mechanism. Later on, it illustrates the abstract syntax (metamodel) and concrete syntax of the proposed DSL “UML-SOA-Sec”. End of the chapter describe in details about the “Saleem’s MDS Services Composition Framework”.

4.1 Proposed DSL: “UML-SOA-Sec”

The MDE uses the concept of the DSL to make it possible to build larger, more accurate, reliable and maintainable software systems [10]. Specification of a DSL that allows the software products to be represented without ambiguity at conceptual levels is one of the most important concerns when elaborating a Model-Driven development solution [103]. To gain the benefits of the DSL and general purpose modelling language, DSLs are defined in terms of general purpose modelling languages like the UML or BPMN [46]. General purpose modelling languages like the UML can easily be customized by the extension mechanism provided by the language itself and the DSL can be defined according to the domain of interest by extending the general purpose modelling language. In case of the UML the extension mechanism is known as the UML Profile. Tools are available for the general purpose modelling languages which support the definition and usage of the DSL.

There is no universal approach for the integration of security and design modelling languages [107, 109]. The current practice of defining a DSL by different researchers in the related work is [4, 8, 44, 56, 93, 101]: abstract syntax of the DSL is represented by a metamodel and the concrete syntax is represented by a UML profile. This work is also working along this approach and defines the abstract syntax of the proposed DSL by a metamodel; and the concrete syntax by a UML profile. The UML profile diagram is created according to the guide-lines mentioned in [206].

Figure 4.1 shows the abstract picture of the whole work. "*Language*" is formalized based on some "*Concepts*" and "*Tool*" is developing to support these "*Language*". This "*Language*" provides the extension mechanisms and "*Tool*" also support the "*Language*" extension mechanisms. This work focuses on the domain of "*security modelling for SOA Applications*" and extends the general purpose modelling language, UML, by providing a metamodel and a UML profile. The proposed DSL is given the name "UML-SOA-Sec". The MagicDraw tool is used for the UML modelling which supports the definition and usage of the DSL.

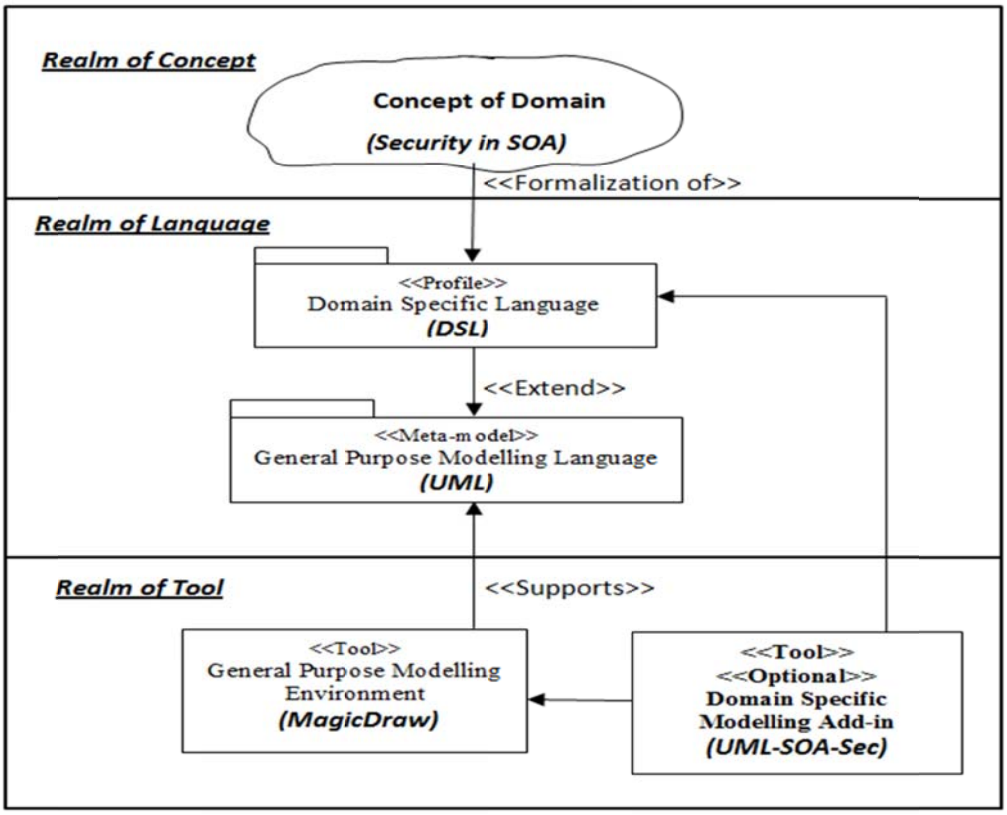


Figure 4.1: Definition Process of a Domain Specific Language [43]

4.1.1 Abstract Syntax of UML-SOA-Sec

In MDE, conceptual elements of a domain are formalized using metamodels at different levels of abstraction. As an MDS, the conceptual elements are formalized at the Platform Independent level where business objectives and security objectives are modelled and integrated using an appropriate modelling technique [10]. The metamodel in this work uses MOF framework for the integration of security objectives along business requirements. Abstract syntax of the proposed DSL "UML-SOA-Sec" is presented by a metamodel which defines the basic concepts, their relationship and the integrity constraints [102]. The UML profile that describes this metamodel is presented as UML package with the stereotype <<profile>> as shown in Figure 4.2. The package is used for the creation of the DSL in accordance with the OMG's specification [98].

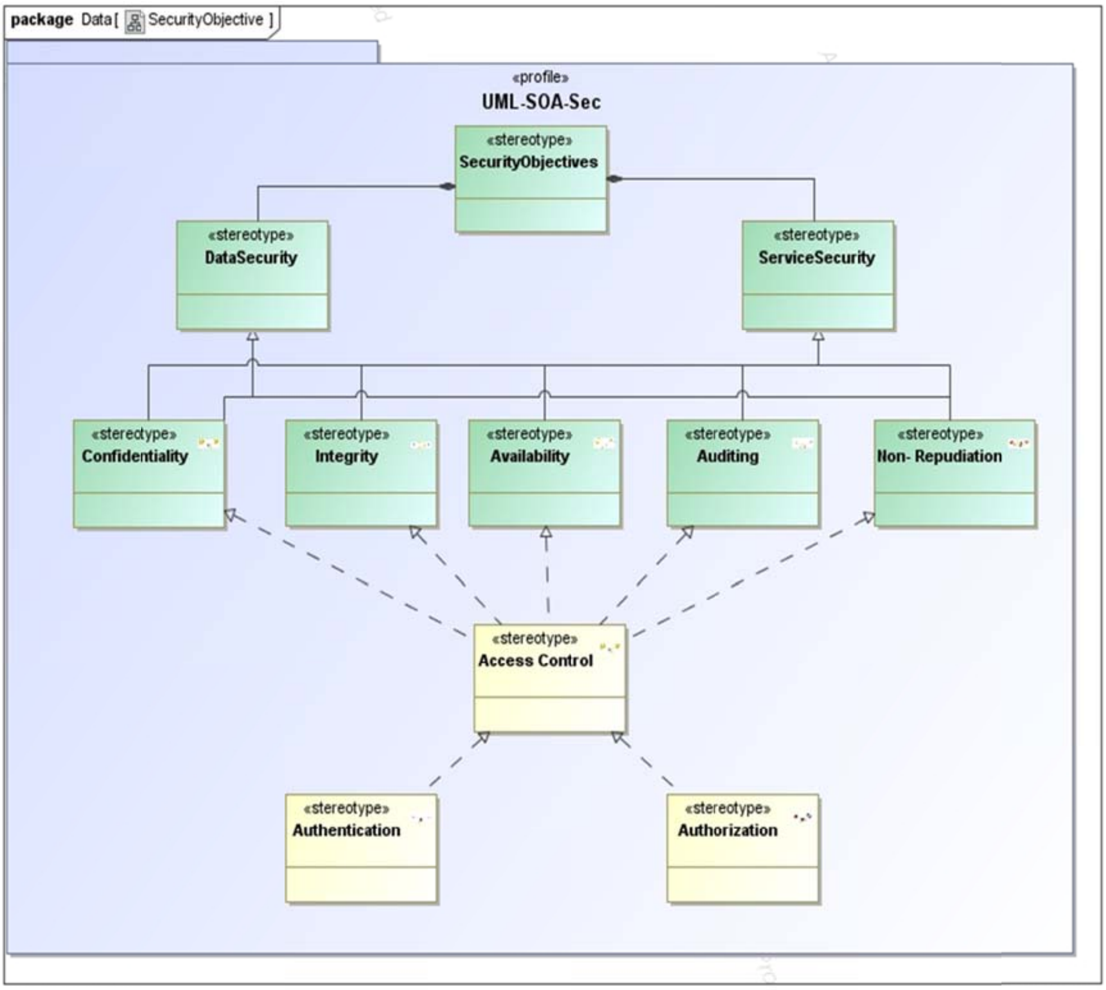


Figure 4.2: Abstract Syntax of Proposed DSL UML-SOA-Sec

Figure 4.2 represents two things, security objectives and security mechanisms. The first three rows are the security objectives and the last two rows are the security mechanisms through which these security objectives can be realized.

In the proposed DSL "UML-SOA-Sec", security objectives are defined for data security as well as service security because in the SOA environment both, data as well as services need to be secure, that's why "*Securityobjectives*" stereotype is composed of two stereotypes "*DataSecurity*" and "*ServiceSecurity*" "DataSecurity" and "ServiceSecurity" stereotypes are generalizations of the five stereotypes namely Confidentiality, Integrity, Availability, Auditing and Non-Repudiation, which are the essential security objectives of an SOA environment. These five stereotypes are realized through the Access Control stereotype, which itself is realized through the Authentication and Authorization stereotypes that are the security mechanisms to achieve the security objectives.

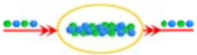
The most difficult task is the identification of elements of the metamodel of a modelling language, which would be extended for the definition of a DSL e.g. in the case of the UML, the identification of UML meta-classes for which the stereotypes will be defined [43]. In this work, the UML meta-classes, O*bjectNode and ActivityNode,* are extended i.e. these are the meta-classes to which stereotypes will be assigned.

4.1.2 Concrete Syntax of UML-SOA-Sec

The concrete syntax of a DSL defines the notion of the language which will be used during modelling i.e. the front end of the DSL. Generally, there are two possibilities to define a concrete syntax (notion) for the elements (in the present case, security objective) of the metamodels of a DSL. The first option is to express them as a property of the subject of the element. The second option is the definition of artifacts for each element that can be used to annotate the subject of the element e.g. an Activity in a UML Activity diagram [107]. The second approach is used during this work to define the concrete syntax of the proposed DSL "UML-SOA-Sec".

In the proposed DSL, each extension of the elements of the UML metamodel is formally captured under the concept of "*stereotypes"*. Properties and/or modelling constraints of the target domain are associated with the stereotypes which results in the "*UML profile"*. Table 4.1 represents the concrete syntax of the proposed DSL. Meaningful symbols (icons) have also been proposed for the security objectives as well as security mechanisms.

Table 4.1: Concrete Syntax (Notions) of the Proposed DSL

S/No	Stereotype Name	UML-Meta Classes	Symbols	Description
1	Confidentiality	ObjectNode ActivityNode		It is applied on the activity as well as on the data communicated to represent that they are private.
2	Integrity	ObjectNode ActivityNode		It is applied on the activity as well as on the data communicated to represent the integrity of the service as well as the data.
3	Availability	ObjectNode ActivityNode		It is used to represent the availability of a data or service.
4	Non-repudiation	ObjectNode ActivityNode		It is used to represent that the data and service usage contain the information about the usage by applying a digital signature.
5	Auditing	ObjectNode ActivityNode		When this stereotype is applied, it represents the auditing of some action.
6	Authentication	ObjectNode ActivityNode		It is applied on a party who wants to initiate a collaboration
7	Authorization	ObjectNode ActivityNode		This stereotype is used to represent the communication between the two parties where the requester has to go through the authorization check.

After the definition of the domain specific UML-profile, a general-purpose modelling tool can easily be specialized and these domain specific stereotypes are made available at the modelling level in the form of annotations [43]. During this research work, the MagicDraw UML modelling tool is used which supports the definition of stereotypes, assignment of icons with the stereotypes and usage of stereotypes in a UML model.

4.1.3 Discussion

From its inception, the UML was designed to be customizable and it provides many semantic variation points in which the UML specification either explicitly or implicitly allows for multiple possible interpretations. The requirements of modelling a specific domain can be fulfilled by elements of the UML by just restricting the UML elements, adding syntactic sugar or adding some constraints to them while respecting the original semantics of these UML elements. These extensions are grouped in the form of a UML profile. The UML also provides special language constructs for refinement and stereotypes are one of them [101].

Stereotyping of the UML model elements is a convenient way to identify those semantics of modelling elements which go beyond the confines of the UML standards. A stereotype is basically a sub-class that refines its base meta-class. The question may arise: instead of defining stereotype, why not just create a new element in the UML metamodel in a standard way? There are two main reasons why people are not moving in that direction [101].

1. *Allow the flexibility to the tool implementer in choosing their preferred approach:* When a new element is added in the metamodel of a modelling language, it implies that new stereotypes can also be added; this result in the modelling tool being a meta-CASE tool i.e. the tool dynamically modifies the modelling language that it supports. Some modelling tools are developed allowing this to happen e.g. the MagicDraw UML modelling tool. However, other tools usually encode the rules and constraints defined by a modelling language in the program code. It implies that to change the metamodel, it requires re-programming the modelling tool.

2. *The need to support different viewpoint type profiles which are dynamically able to apply and un-apply a stereotype during modelling:* When a stereotype is applied to a modelling element, an attachment is linked to that modelling element containing the information about the applied stereotype and any values associated with its

attributes. When the stereotype is un-applied then this attachment is removed from the modelling element. Sometimes stereotypes also contain OCL constraints used to capture the domain-specific constraints. These OCL constraints are applied to stereotypes but not to their base classes.

4.2 Proposed Saleem's MDS Services Composition Framework

Saleem's MDS services Composition framework is proposed for the model-driven development of a secure web services composition. In the proposed Saleem's MDS services composition framework; the four most important steps are selected among those steps discussed by different researchers. The main contribution is the definition of security objectives along business process modelling, which is performed using the UML activity diagram at two different stages i.e. at step-1 and step 3. The proposed DSL "UML-SOA-Sec" is used for security modelling. The proposed framework identifies the steps where the business process expert has to model the security objectives along the business process modelling for services composition. This security annotated business process model will be transformed into a services composition using a model-driven approach. Now the focus is to compose an advanced service (services composition) out of basic atomic services.

The principle of the MDS services composition framework is to use the UML to model the services composition and from the UML models, to generate the BPEL model and WSDL specification. In the area of web services composition research, there are two main aspects that are modelled: the service and the workflow. Service modelling identifies services to be exposed with their interfaces and operations, (a UML Class diagram is used for service modelling) while the workflow modelling identifies the control and data flows from one service to the next (a UML activity diagram is used for workflow modelling) [23].

A *web service* is represented by UML class diagram, which may consist of several operations that are represented by the operations of the UML class diagram. Each individual operation has its own internal behavior, which is expressed by the UML activity diagram. The UML activity diagram prescribes the implementation of the operations with the help of a structured set of activities called *composition*, which describes the control flow and data flow among the activities. An activity in the UML activity diagram corresponds to the invocation of a web service operation. A composite web service has one or more operations which invoke basic web services [24].

The goal of the proposed framework is to design the secure web services composition using the UML class diagram and UML activity diagram. Many researchers have presented several steps/phases for their web services composition framework/method like Roy Grønmo and Ida Solheim [23], Christophe Dumez et al. [51, 85], Skogan et al [24], Jun Han et al. [3] and Andre R.R. et al.[47] etc. In the proposed MDS Services Composition framework the four most essential steps of services composition are selected and organized. The most prominent contribution of this research work is the defining of the security in the UML activity diagram at step-1 and at step-3 with the help of the proposed DSL "UML-SOA-Sec".

The following is a detailed description of the proposed model driven security framework for services composition as described by Figure 4.3.

4.2.1 Step 1: UML Modelling of Services Composition:

The goal of this step is to perform modelling for the composite application and to identify the candidate web services. At the preliminary stage, two kinds of modelling are performed: firstly, the UML class diagram is modelled to describe the interface of the proposed composite web service. Secondly, the UML Activity diagram is modelled to show the sequence of operations to be performed and the control flow of the proposed services composition. The security objectives in UML Activity diagram are incorporated by using the "UML-SOA-Sec", which will result in a security annotated UML activity diagram. Web services are identified by their names and textual descriptions in the UDDI registry, but their operations are described in the WSDL files. The business process expert (modeller) has to search, discover and extract the WSDL description of the web services from the web service repository, which is placed on the Internet server.

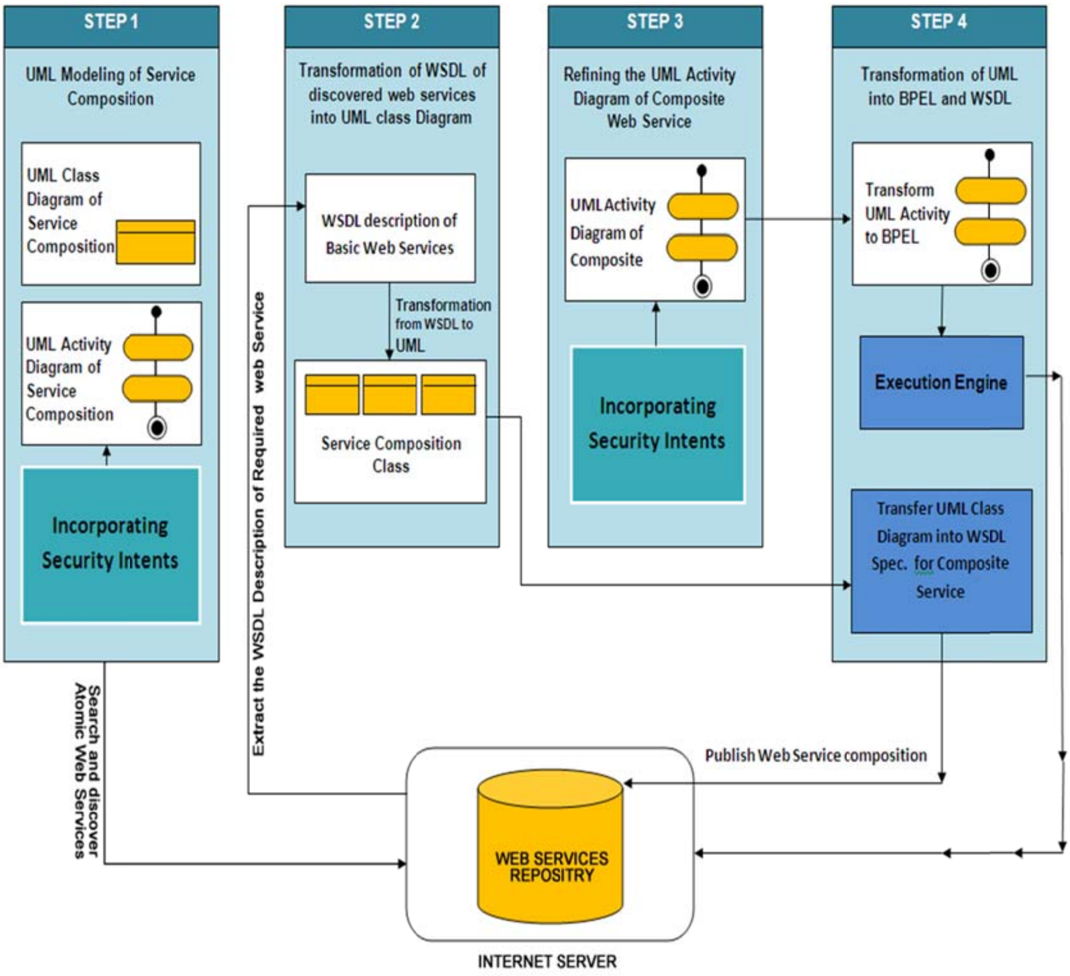

STEP 1
UML Modeling of Service Composition
UML Class Diagram of Service Composition
UML Activity Diagram of Service Composition
Incorporating Security Intents
STEP 2
Transformation of WSDL of discovered web services into UML class Diagram
WSDL description of Basic Web Services
Transformation from WSDL to UML
Service Composition Class
STEP 3
Refining the UML Activity Diagram of Composite Web Service
UML Activity Diagram of Composite
Incorporating Security Intents
STEP 4
Transformation of UML into BPEL and WSDL
Transform UML Activity to BPEL
Execution Engine
Transfer UML Class Diagram into WSDL Spec. for Composite Service
Search and discover Atomic Web Services
Extract the WSDL Description of Required web Service
Publish Web Service composition
WEB SERVICES REPOSITRY
INTERNET SERVER

Figure 4.3 Framework for Model Driven Development of Secure Web Services Composition

4.2.2 Step 2: Transforming of the WSDL of the discovered Web Services into a UML Class Diagram

The goal of the second step is to model the details of the services composition. The business process expert (modeller) transforms the WSDL description of the extracted web services into a UML class diagram. At this stage, the basic services are identified that are used to generate the UML class diagram of the services composition which will serve as the interface for composite web service.

4.2.3 Step 3: Refining the UML Activity Diagram of the Composite Web Service

During this step, refining of the UML Activity Diagram of the composite service is achieved by adding the details found in the WSDL description of the web services. It will define the internal behavior of the services composition. Refinement of the security objectives is also achieved and preparation of a security enhanced UML activity diagram for the services composition.

4.2.4 Step 4: Transforming of UML Models into a WSDL and a BPEL

During this step, UML models are transformed into code. At the end of this step, the composed service is ready for deployment. The UML Activity diagram will be transformed into an executable specification like the BPEL and its specification will be deployed on a work flow engine which produces an implementation code for handling control-flow and data-flow. The UML Class diagram is transformed into a WSDL description which would be published in the Web Service repository. Once the web service is published in the registry, it can be discovered and used by the web service consumers.

4.2.5 Discussion

The main contribution in the proposed MDS services composition framework is incorporating security objectives for modelling of the services composition. In the proposed framework, security is defined at two different stages. Firstly, at step-1, the overall modelling of the services composition is performed using the UML activity diagram. This is the concept building stage about services composition i.e. what functionality this services composition has to perform, which services are required to accomplish this functionality and what are the security requirements for this composed service. Secondly, at step-3, when all required services are either discovered or developed; now all the required services are available and security will be refined/redefined for modelling of the services composition using the proposed DLS "UML-SOA-Sec" in the UML activity diagram

4.3 Chapter Summary

This chapter starts with illustrating the detail about the proposed DSL “UML-SOA-Sec”. Detail discussion is presented about the abstract syntax and concrete syntax of the “UML-SOA-Sec”. At the end the chapter, a detailed discussion is presented about the proposed “Saleem’s MDS services composition framework”.

CHAPTER 5
SECURITY ENHANCED DESIGNING OF AN SOA APPLICATION

5.0 Chapter Overview

This chapter is dedicated to proof of concepts designing of the proposed work. This thesis claims contribution in two areas, i.e., Security Modelling in the Business Process Modelling for SOA applications and secure Web Services Composition. In this chapter, the proposed DSL "UML-SOA-Sec" is validated by applying it on a business process model of a real world example of a healthcare system.

5.1 Introduction of Healthcare System

In today's internet-based vulnerable Information System, information security is an open debate. This system may be banks, healthcare, social networks, or some e-commerce application. For healthcare system, due to the privacy requirements of citizen's medical data, security challenges are multi-faceted and it is one of the frequently debated issue all over the world in the government bodies, legislative communities, healthcare industry, and healthcare administration [207, 208]

For healthcare systems, paper based record keeping has been in usage for a long time, in which information is kept in a dispersed form without any correlation between the available data. The intensity of the issue increases when the patient is moved to some other hospital and is asked to bring the "*medical history*" which is normally kept with the previous hospital or sometime the patient may be asked to undergo medical tests which he/she has already gone through. Major disadvantages of the paper based record keeping are the difficult and possibly painful process for the patient, high costs, time consumptions and even possible, medical errors [144].

Like other services industries, healthcare is also integrated with ICT (Information and Communication Technology) [209]. Electronic healthcare systems got started with a vision of storing and using healthcare data in an organized manner. Several standards are defined for the integration of Medical Technology and Information System integration [210, 211]. "*As Information Technology increasingly penetrates the healthcare industry, physicians and patients are experiencing the benefits of on-demand access to medical information where, when and how it is needed*" [212].

5.2 Importance of Security for Healthcare Systems

The primary purpose of medical record storage and retrieval is to provide timely healthcare service to citizens. The medical record of a citizen contains two types of information. The first type is related to the identity of the citizen e.g. name, social security number, address, date of birth, telephone etc. While the second type is medical history of the citizen e.g. information about allergies, diseases, diagnostics tests, radiography images etc. Different people in the healthcare organizations access the medical data e.g. general physician, specialist, radiography specialist, pharmacist, insurance company personals etc. [139]. Security and privacy of the citizen's medical record is a very sensitive issue. The need for privacy is in the nature of human beings and a medical record is considered as the private information of a patient. A person may have a psychological, mental, physical, sexual or emotional disorder which he/she doesn't want to share with other people and they fear that their abnormality would be exposed to others including colleagues, friends and family [139]. The founder of the Patient Privacy Rights says in his article [213] published in the Wallstreet Journal: *"Electronic record systems that don't put patients in control of data or have inadequate security create huge opportunities for the theft, misuse and sale of personal health information"*

5.3 Security Objectives of Healthcare Systems

Healthcare data is stored with a healthcare partner e.g. diagnostic lab, clinic, pharmacy, insurance company etc.; therefore providing security and privacy for the medical data is the key responsibility of healthcare applications. For better care and appropriate diagnostic treatment, timely access to the medical data by the authorized user is very essential [139]. The system should prevent the disclosure of medical data to un-authorized users who could exploit it for criminal or commercial use. Therefore confidentiality and integrity of the data should be ensured by the system, which is technically ensured through proper access control mechanism. Access control itself is ensured through "*authentication*" and "a*uthorization*" mechanisms. Partner organizations make sure that only authenticated and authorized users with specified permission would get access to the medical data. In a simple scenario different users in healthcare organizations like the primary physician, medical specialist, radiologist, insurance personal etc. are assigned certain roles based on their organizational responsibilities to access the medical data. These roles are assigned certain credentials through which they are identified and after proper authentication and authorization checks they are allowed to use healthcare data and services.

Healthcare scenarios involve the sending and receiving of several documents. It is important that, once a particular document is sent or received; one should not be able to deny having sent or received the document. If a user accesses the data or a service, then for evidence, the system should log the details of this access event; therefore, accountability of accessing the data is also an important security requirement which is achieved through a "*non-repudiation*" mechanism which create signed evidence for accessing the system resources [139]. Non-repudiation ensures accountability regarding access to healthcare documents and services.

Keeping in mind the security objectives of healthcare systems (i.e. confidentiality, integrity and non-repudiation), this research work proceeds with the following specific case-study of a healthcare scenario.

5.4 Application of Proposed Work to the Healthcare Scenario

The SOA architectural style is suitable for the healthcare industry where information is integrated and shared among heterogeneous entities which are disseminated far apart and changes occurs frequently. The SOA provides many features to a healthcare environment like integration, availability, reusability, efficient information management and interoperate-ability etc. [144, 214, 215].

In the SOA environment, a system is composed of services offered by different partners. These services from different partner organizations are integrated/composed to form application and every organization work independently without any central control [182]. The distributed nature of the SOA environment and technology-heterogeneity of the services, raise many security challenges. The inter-organizational workflow in the SOA environment is executed in a decentralized manner where one need to secure local data stores as well as the communication among the partners [139]. In such environments different use-cases can be derived from the scenarios to model the business process to an SOA-based system, which leverages the Service-Oriented Analysis and Design (SOAD) methods to be able to determine functional areas that need to be transformed into services [183, 184]; afterwards security objectives would be defined in the business process model.

Realizing the importance of security objectives for healthcare systems, the proposed DSL *"UML-SOA-Sec"* is applied to define the security objectives during the business process modelling. Afterwards, the proposed MDS services composition framework is used for the generation of a secure services composition. The proposed DSL and proposed MDS services composition framework are general and cater for the requirements of different service-oriented domains like e-government, e-health, e-education etc. Furthermore, the security objectives like authentication and authorization are principally the same for different domains [216].

For demonstration of this research work, the SOA based healthcare scenario is used which is a Healthcare Service Specification Project (HSSP) specified service called the Entity Identification Service (EIS) [185]. HSSP is a joint effort by Health Life 7 (HL7) and OMG. The objective of HSSP is to provide baseline for service based healthcare environment and it provides service interface specifications; however, it does not provide information about the implementation. So designing the services using HSSP guidelines and implementing them in one's own way, gives flexibility in the realization of healthcare SOA. The vision of HSPP is to make healthcare free from multiple vendors and give "all-in-one" solution that fits to overall healthcare environment [185].

For the EIS, the basic services are well defined and already modelled using the UML activity diagram [144] and the UML class diagram [144]. This work incorporate security in the UML activity diagram using the proposed DSL "UML-SOA-Sec"; afterwards, it implements security in the composite application.

5.5 Case Study: Security Enhanced Business Process Modelling of Healthcare System

The scenario presented in Figure 5.1 depicts the interaction between two healthcare organizations the "*Sample Collection Point*" and the "*Medical Test Centers*". Users from these two organizations access resources i.e. services and documents within and from other organizations.

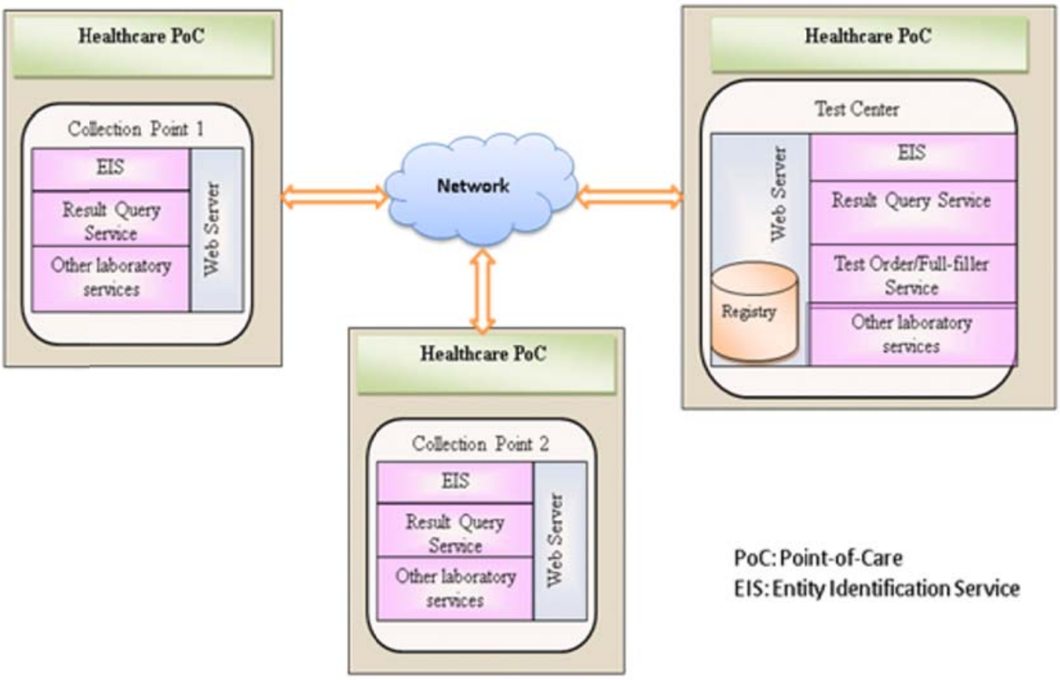


Figure 5.1: System Architecture of the Healthcare System [144]

The healthcare system under study is an SOA environment where services are deployed on different servers at different sites. Each site contains its own database and performs both roles, i.e. service provider as well as service consumer. Healthcare partners create, store, send and receive different kinds of healthcare documents such as patient's information, patient's test results, query result information etc.

5.5.1 Identification of Security Objectives

Based on the scenario discussed where communication takes places among different healthcare partners having an SOA environment, a number of security objectives can be identified. The system should prevent the disclosure of medical data to un-authorized users who could exploit it for criminal or commercial use. Therefore, *Confidentiality* and *Integrity* of the data should be ensured by the system; it is technically ensured through proper *Access Control* mechanisms. Access Control itself is ensured through "*Authentication*" and "*Authorization*" mechanisms.

A patient visits a "*Sample Collection Point*" and provides his/his demographics and asks for the medical information or any test. The person sitting at the collection point, before interacting with the healthcare system has to prove his/her identity which enables him/her to get an appropriate role in the system and grant him/her permissions based on his/her role. The same is the case with the person sitting in the "*Medical Test Center*", when he/she has to interact with the system he/she has to prove his/her identity. The identity validation is performed through an *authentication* mechanism and role/permission verification is performed through an *authorization* mechanism [139]. The system at the sample collection point and the test center validate the person's credentials representing his/her identity for granting him/her access to the services offered by the system. These credentials could be a username-password or a digital certificate or any other kind of access control mechanism.

In the same context when the "*test order*" document is sent from the sample collection point to the test center or the "*test result*" document is sent from the test center to the sample collection point; users from both places should not be able to deny having participated in the interaction. The security requirement for accountability and auditing is called *non-repudiation*. Basically it is used for the access and usage of resources and if it is ensured, then it is not possible for users to deny having accessed the system resources; therefore misuse of medical data can be avoided [139].

Having discussed the security objectives in the general healthcare scenario, the following is a description of the specific scenario which is taken as a case study for this work.

5.5.2 Entity Identification Service (EIS).

The EIS is a HSSP specified service which lies under the Patient Administration Domain [144]. Basically, the EIS focuses on the patient's demographics and is responsible for the retrieval of patient identification information [185]. EIS service itself is the outcome of a composition of multiple services with one particular business goal. The architectural design of this service is according to the HL7 V3 specification [144]. The static structure of the EIS is represented by the UML class diagram in Figure 5.2 [217].

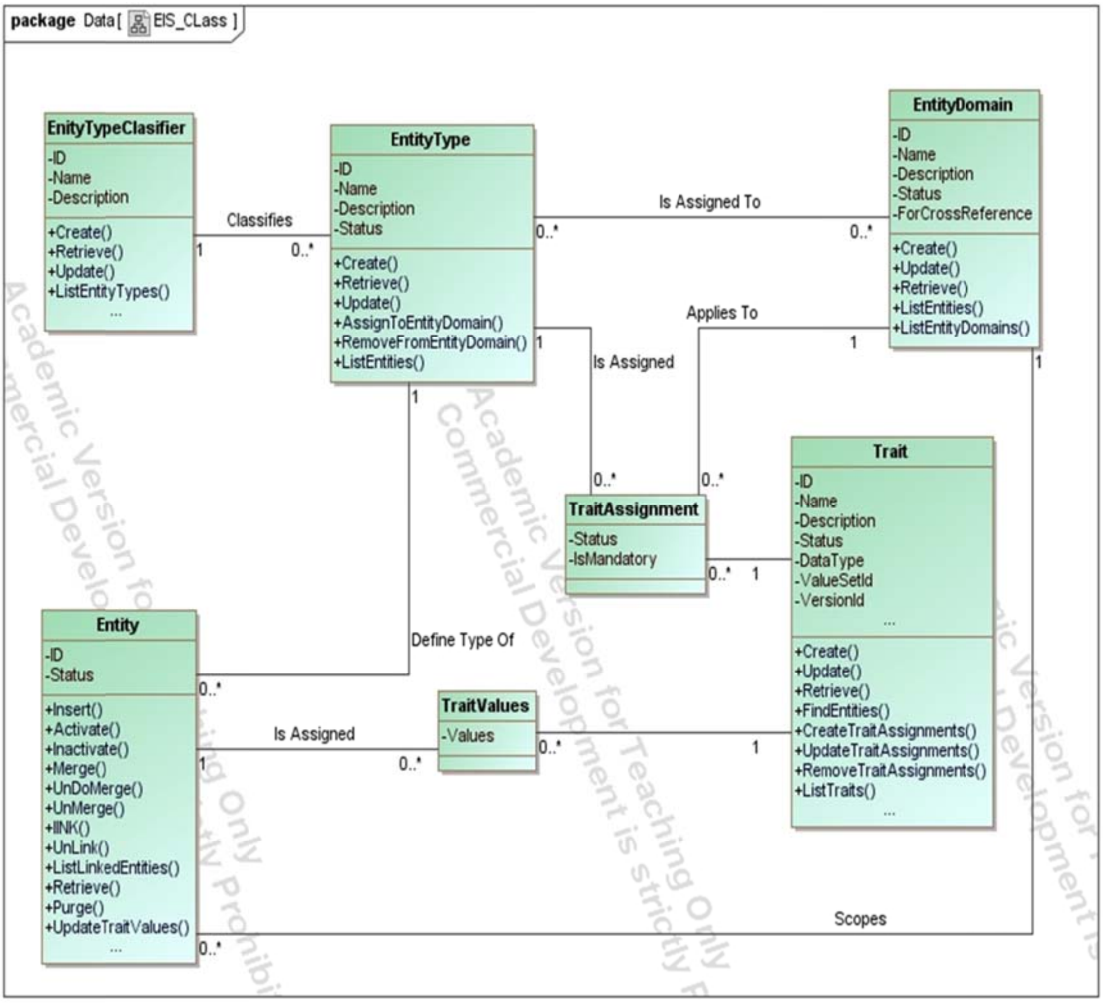


Figure 5.2: UML Class Diagram of EIS

The EIS consists of several operations which are represented by the operations of the UML class diagram. Each individual operation of the EIS has its internal behavior which is expressed by the UML activity diagram. The UML activity diagram prescribes the implementation of the operations with the help of a structured set of activities called a *composition*, which describes the control flow and data flow among the activities. An activity in the UML activity diagram corresponds to the invocation of a Web service operation. A composite Web service has one or more operations, which invoke basic Web services. In this work, the "*retrieve"* operation is taken and the UML activity diagram is modelled based on it.

5.5.3 Retrieving Patient's Demographic in EIS

The work flow of "*retrieve*" operations is discussed in this section. A patient visits the sample collection point and asks for medical information or any test. He/she provides his/her demographics; first his/her record is searched for at the local database. If the record of the patient does not exist on the local collection point then the query is sent to the external system which may be another collection point or a test center. If the patient's record is found there then it is sent to the originated site. If the patient's record also does not exist on the external system, then a new patient profile is created. The EIS is composed of several services whose objective is to identify the patient locally or remotely. This paragraph provides the description of their workings. The EIS entry-point-service is basically an entry point to the composite service. The client application wants to search the record for the desired patient by providing the "Patient_ID". Before accessing the composite application, the "*Confidentiality*" and "*Integrity*" security objectives must be ensured. After ensuring these two security objectives, the "Patient_ID" information is forward to the Message Generator Service, which takes the value and generates the required "*Patient Registry Get Identified Query*". The generated message is sent to the Parser Service; where it is parsed. The parsed message is then sent to the Database Service. The business logic of the Database Service consists of the Patient Demographic Retrieving Functions. If the patient record is found locally, then it is sent to the client application; otherwise, the same request is sent to the External EIS (XEIS), which itself is the same kind of EIS business logic installed at some other location. The "*Non-repudiation"* security objective is ensured in this type of communication.

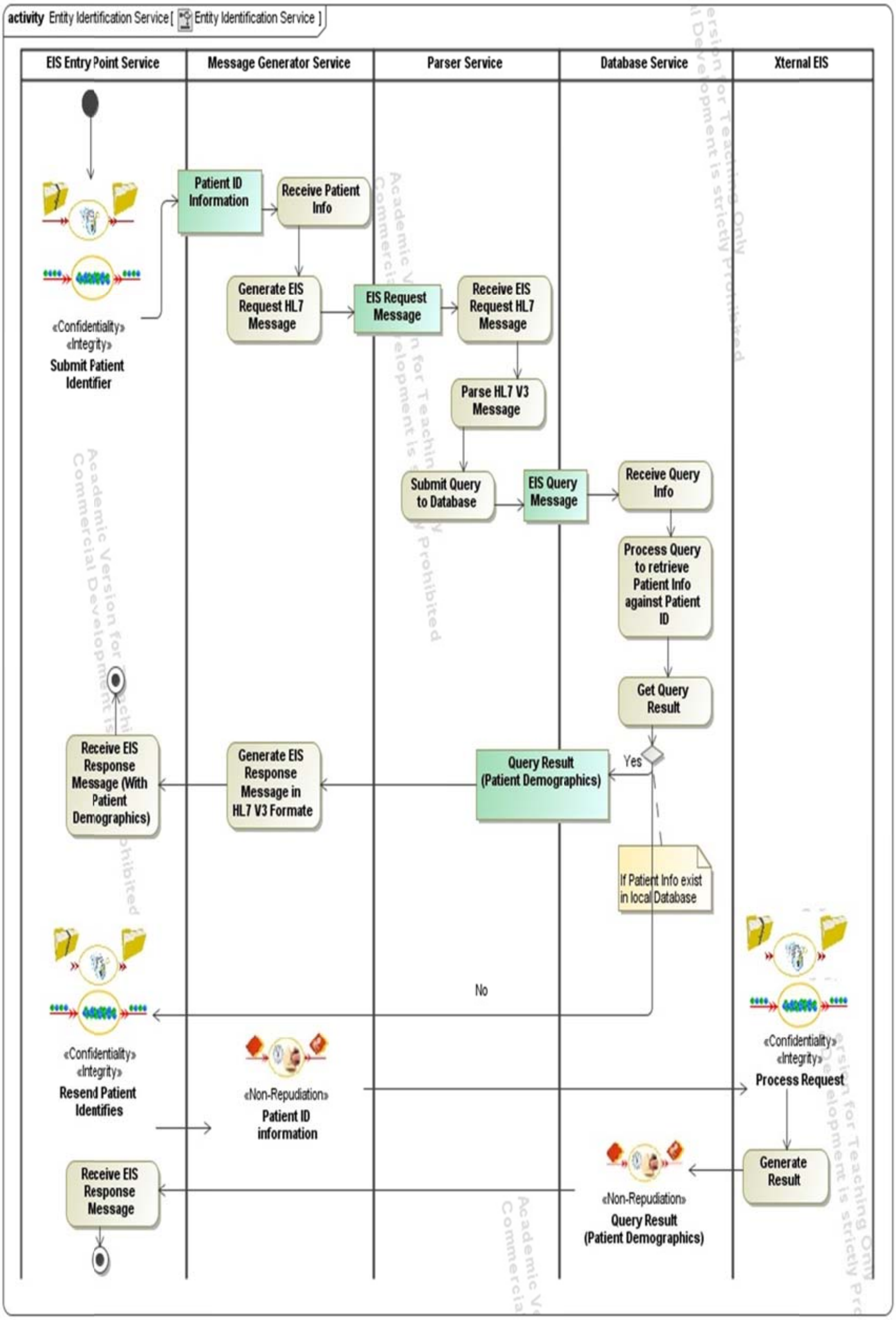

activity Entity Identification Service [Entity Identification Service]
EIS Entry Point Service
Message Generator Service
Parser Service
Database Service
Xternal EIS
«Confidentiality»
«Integrity»
Submit Patient Identifier
Patient ID Information
Receive Patient Info
Generate EIS Request HL7 Message
EIS Request Message
Receive EIS Request HL7 Message
Parse HL7 V3 Message
Submit Query to Database
EIS Query Message
Receive Query Info
Process Query to retrieve Patient Info against Patient ID
Get Query Result
Yes
Query Result (Patient Demographics)
Generate EIS Response Message in HL7 V3 Formate
Receive EIS Response Message (With Patient Demographics)
If Patient Info exist in local Database
No
«Confidentiality»
«Integrity»
Resend Patient Identifies
«Non-Repudiation»
Patient ID information
«Confidentiality»
«Integrity»
Process Request
Generate Result
«Non-Repudiation»
Query Result (Patient Demographics)
Receive EIS Response Message

Figure 5.3: Security Annotated Entity Identification Service

The security annotated workflow of the "*retrieve*" operation of the EIS is represented through the UML activity diagram where security is modeled using the proposed DSL "UML-SOA-Sec" and security objectives are displayed in the model as text as well as icon, and are shown in Figure 5.3.

5.6 Chapter Summary

This chapter presented an overview of the healthcare systems, the importance of security in healthcare environment and the security objectives of healthcare systems. Afterwards, is described in detail the case-study which was used for the prototype implementation of this research work. Later on, it presents the specific scenario which is taken for the implementation of proposed work; its services were represented by the UML class diagram. The workflow was represented by UML Activity diagrams and the security objectives were modelled in the business process model with the help of proposed DSL "UML-SOA-Sec".

CHAPTER 6
SECURE COMPOSITE APPLICATION DEVELOPMENT

6.0 Chapter Overview

This chapter is dedicated to proof of concepts implementation of the proposed work. This thesis claims contribution in two areas, i.e., Security Modelling in the Business Process Modelling for SOA applications and secure Web Services Composition. In this chapter, the second contribution is validated with a prototype implementation of a secure Web services composition using open source tools and technologies.

6.1 UML Deployment Diagram

The EIS composite application is comprised of business services as well as security services which are deployed on the application server. A Test_Application is also deployed on the same application server, which is basically an interface for the Client Applications to authenticate and pass the session information to the client application. A client may be any application which wants to access the EIS composite assembly to perform a desired operation, i.e. to retrieve patient information. Figure 6.1 illustrates the deployment diagram of the prototype.

Below is provided details about workings of the system:

The Client application uses the web browser to access the Test_Application which uses the Security Service to *authenticate* the client application and retrieves its Session ID (SID). Using this SID, the Client Application performs its operations on the Business Services, which ensure the *authorization* and *non-repudiation* security checks by contacting the Security Service against the particular SID of the Client application. After ensuring the proper security checks the client application can access the EIS composite application to get the task completed.

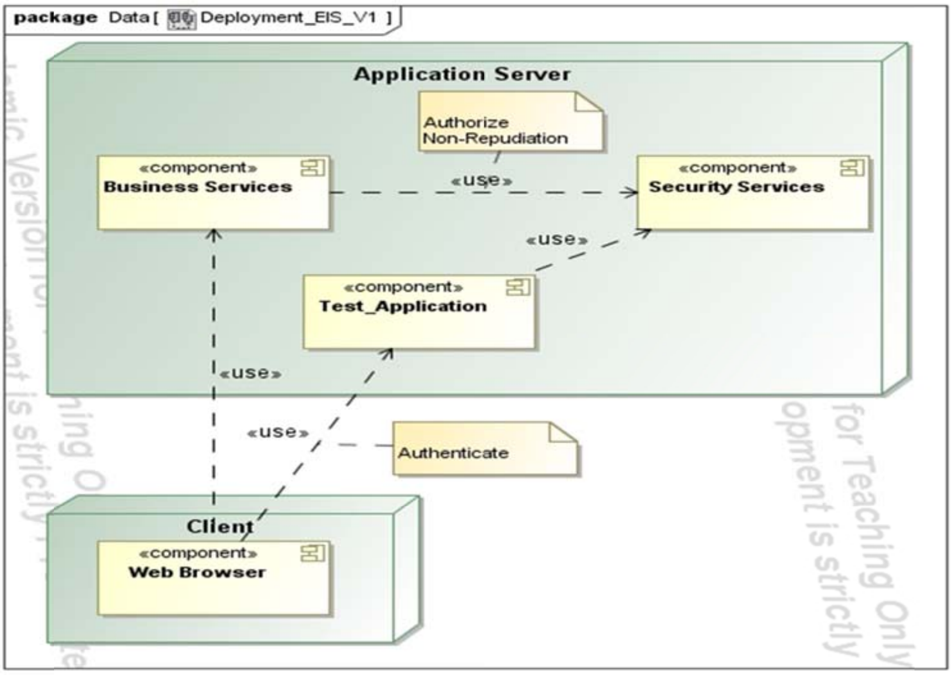


Figure 6.1: UML Deployment Diagram of Prototype Implementation

6.2 EIS BPEL Workflow Diagram

Figure 5.2 shows the BPEL orchestration for the healthcare scenario. EIS is composed of many services whose main functionality is to identify a patient locally or remotely. These services are defined in the WSDL standard, which is basically an XML format for service definition, containing messages. The operations in the WSDL send/receive those messages written in an appropriate message format. During this work, the SOAP message format is used. During this work, these business services are just used as they are already developed [144]. A WSDL file is created which contains the method to call the appropriate service. WSDLs are known as partner links in a BPEL workflow. Web services (WSDL Files), together with the BPEL constructs make overall services composition. In this way BPEL acts as a container for the Web services which are described by their WSDL descriptions.

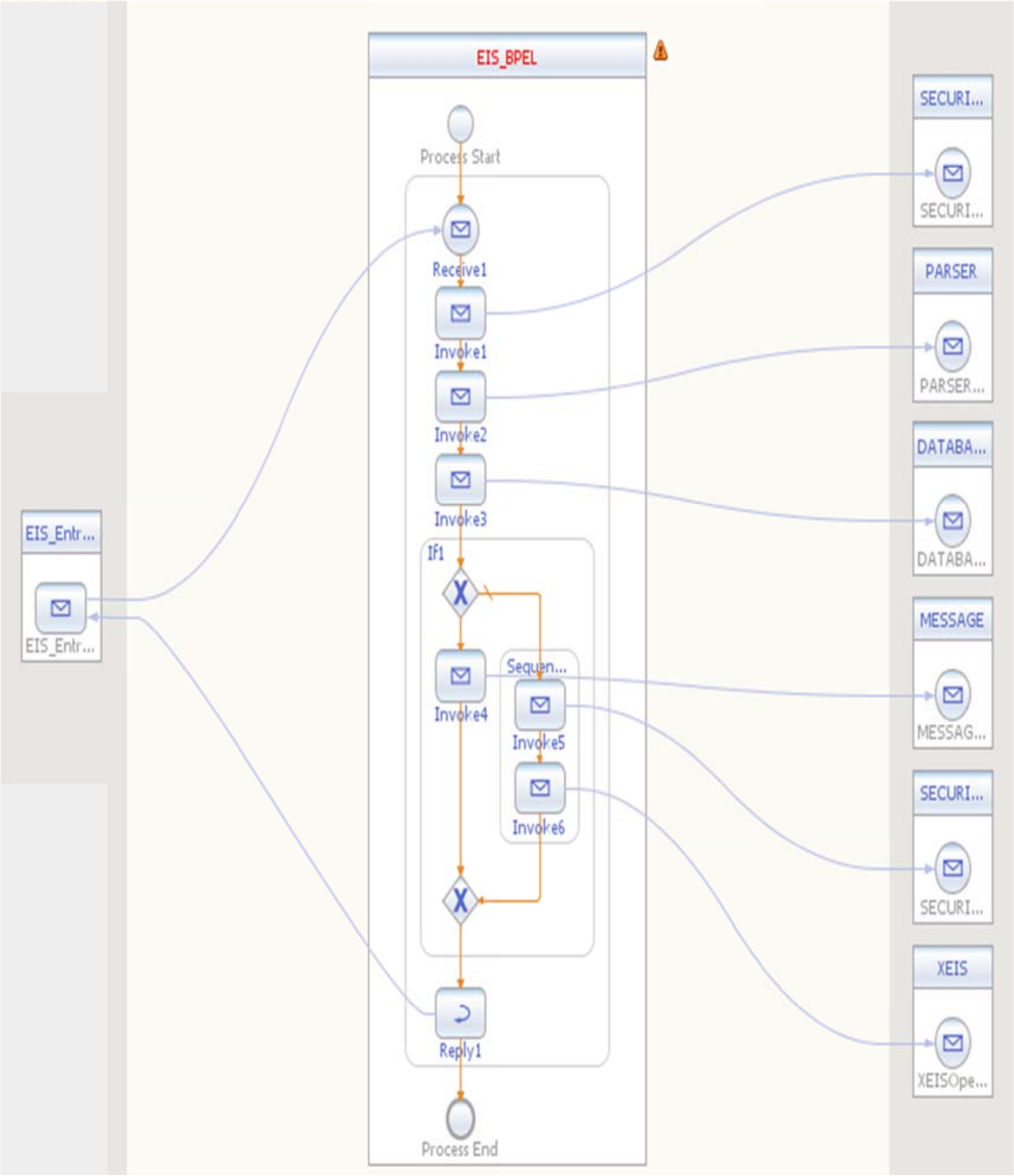


Figure 6.2: EIS Workflow Diagram

Below details are provided about workings of the system:

As can be seen from Figure 6.2, five business services are involved in the EIS workflow; namely: EIS_EntryPoint Service, Message Generator Service, Parser Service, Database Service and External EIS (XEIS). The EIS session starts from the client application which sets the "Patient_ID" as input information. The security objective confidentiality and integrity are ensured through the access control security mechanism. After ensuring security, the "Patient_ID" information are forward to the

Message Generator Service, which takes the “Patient_ID” information and generates the required “*Patient Registry Get Identified Query*”. The generated message is sent to the Parser Service; where it is parsed. The parsed message is then sent to the Database Service. The business logic of the Database Service consists of Patient Demographic Retrieving Functions. If the patient record is found locally, then it is sent to the client application; otherwise, the same request is sent to the XEIS, which itself is the same kind of EIS business logic installed at some other location. The non-repudiation security objective is ensured in this type of communication.

6.3 Description of the EIS Workflow Diagram

The communication among the services along BPEL constructs is presented in Figure 5.2 and illustrated as follows [144]:

1. *The EIS EntryPoint Service* represents a Partner Link, from which the client application sends the request for the patient's demographics against his/her ID to Receive construct.

2. The Receive construct then invokes the *Security Service* which will authenticate and authorize the client application.

3. After authentication, the *Parser Service* is invoked through the Invoke construct.

4. The *Database Service* takes the patient-ID from the parsed message and checks the patient's record against that ID in the local database.

5. If the patient information is found locally, then this information is sent to the *Message Generator Service* to format it according to the HL7 V3. The generated response message is sent to the EIS EntryPoint Service, through the Reply construct.

6. In a case when the patient information is not found in the local database, then the *XEIS* service is invoked in order to check the patient's record at other point-of-care. Before invoking the XEIS, the *non-repudiation* security mechanism is applied so that the communication cannot be denied. Basically, the XEIS

represents the same kind of BPEL workflow deployed on another remote center. If patient's record is found at remote center, it is sent to the EIS EntryPoint service partner link.

The BPEL acts as a container for Web services and it relies on web services interfaces, i.e. WSDL files, which themselves are interfaces for some methods/operations [139]. In this work, these methods/operations are implemented in Java.

6.4 Service Assembly Unit

A JBI environment is used for the integration of the BPEL workflows. Figure 6.3 illustrates the EIS composite application having an assembly of service deployed in the JBI environment.

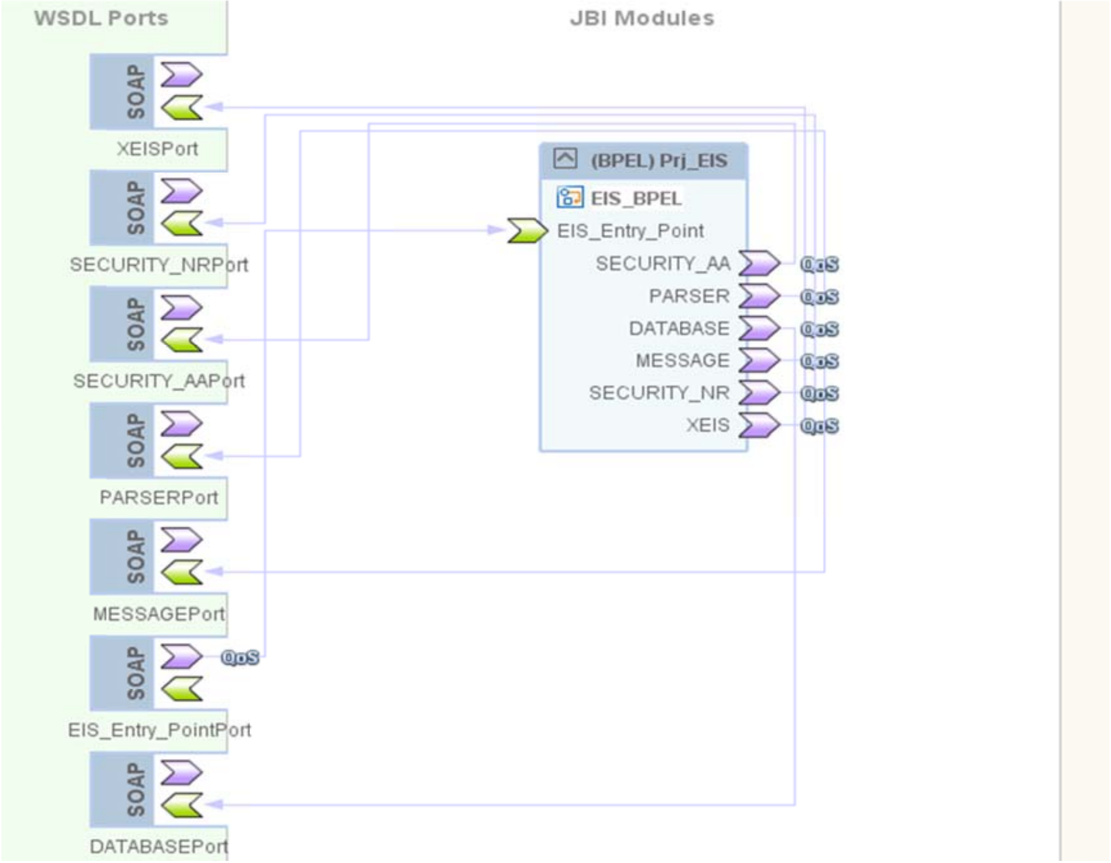


Figure 6.3: EIS Assembly of Service Units

6.5 Composite Application Deployment

After the integration of the BPEL workflow in the JBI environment, the service assembly is deployed in the Tomcat Application Server. During this work, the composite application named the "hl7" and the client application named "hl7c" have been deployed on the same machine; Figure 6.4 describes the deployment description of both.

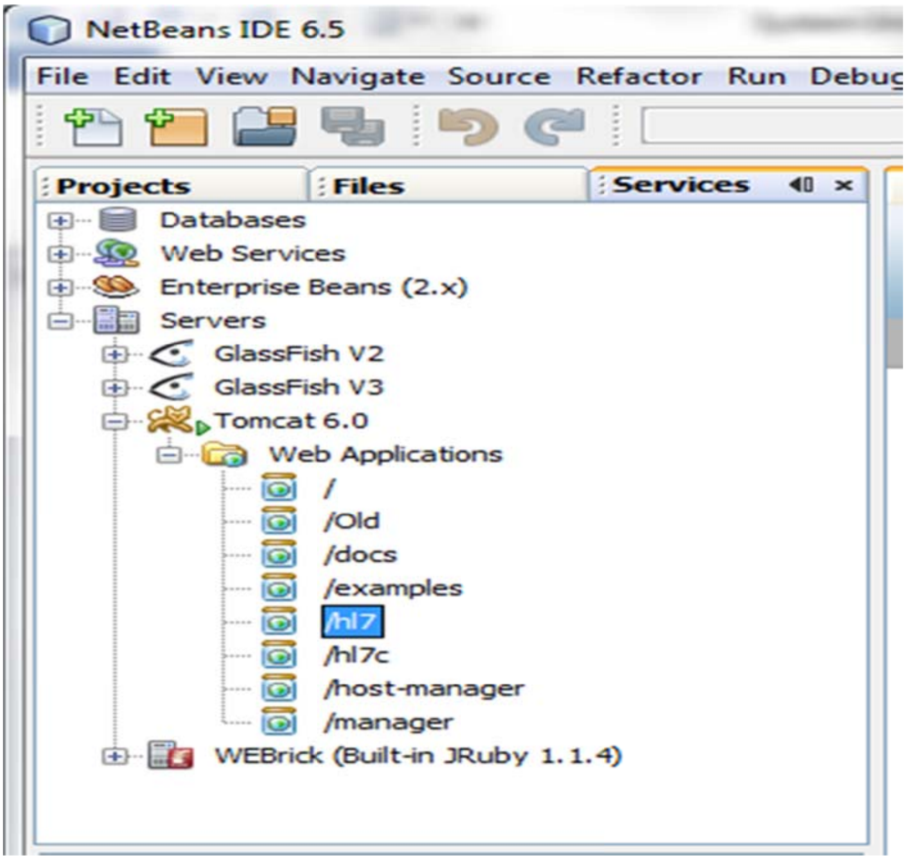
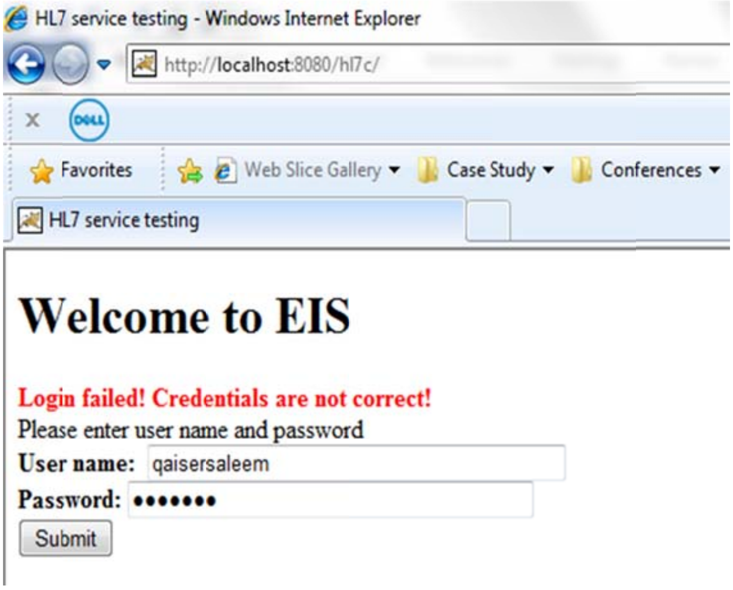


Figure 6.4: Deployment of the EIS Composite Application and Client Application.

6.6 Client Application is Accessing the EIS Composite Application

The EIS composite application is accessed through the client application which wants to perform a desired operation. The client application for the EIS can be a Web application or a Java application, which provides the Web reference of the EIS composite application. Whenever some client application wants to access the composite application, it will provide the security credentials. For the prototype application of this research work, the security credentials used are the "*user-name*" and "*password*" to ensure the security objectives. Figure 6.5 show that when an incorrect user-name and password are provided then the client application is not allowed to access the composite application.


HL7 service testing - Windows Internet Explorer
http://localhost:8080/hl7c/
Favorites
Web Slice Gallery
Case Study
Conferences
HL7 service testing

Welcome to EIS

Login failed! Credentials are not correct!

Please enter user name and password

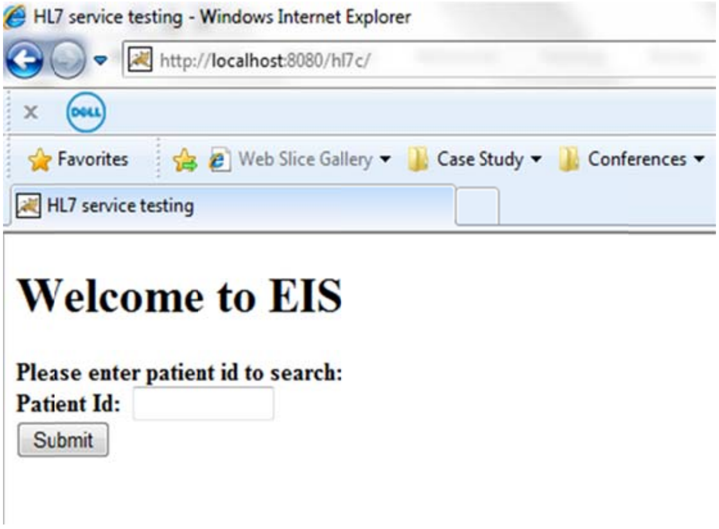
User name: qaisersaleem

Password: •••••••

Submit

Figure 6.5: Incorrect Credentials are Provided for the Username and Password

After providing the correct credentials, the client application can access the composite application and the introductory screen will appear, Figure 6.6, which will take the patient-id as the input to search for the patient's record whether it exists in the database or not.


HL7 service testing - Windows Internet Explorer
http://localhost:8080/hl7c/
Favorites
Web Slice Gallery
Case Study
Conferences
HL7 service testing

Welcome to EIS

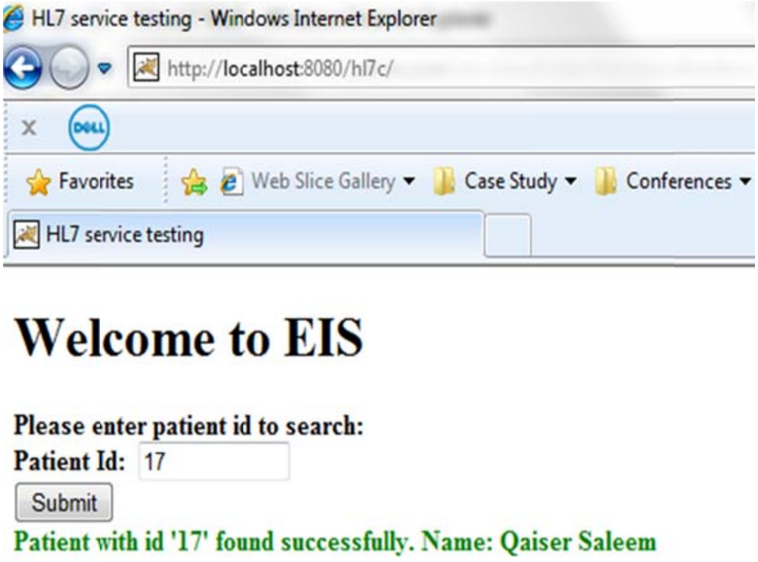
Please enter patient id to search:

Patient Id:

Submit

Figure 6.6 Welcome Screen of EIS to Get Input for Patient-Id

Figure 6.7 and Figure 6.8 illustrate the scenarios when the patient record is found in the database and when the record is not found in the database, respectively.


HL7 service testing - Windows Internet Explorer
http://localhost:8080/hl7c/
Favorites
Web Slice Gallery
Case Study
Conferences
HL7 service testing

Welcome to EIS

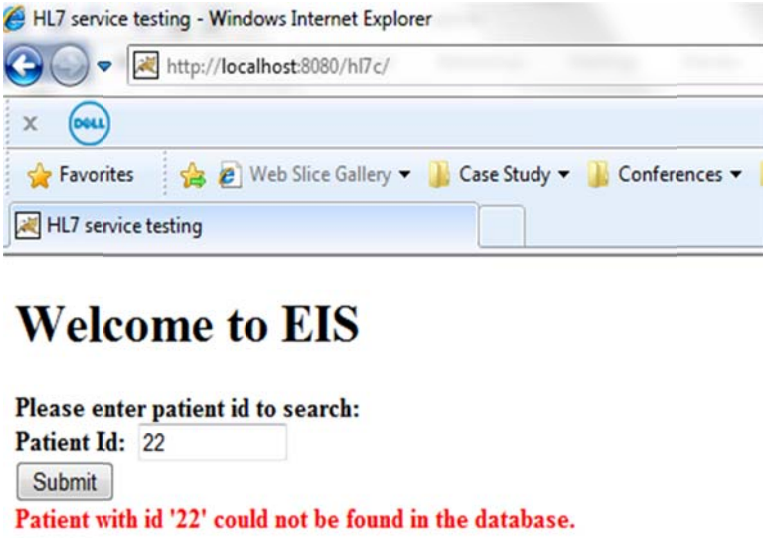
Please enter patient id to search:

Patient Id: 17

Submit

Patient with id '17' found successfully. Name: Qaiser Saleem

Figure 6.7 Patient Record Found


HL7 service testing - Windows Internet Explorer
http://localhost:8080/hl7c/
Favorites
Web Slice Gallery
Case Study
Conferences
HL7 service testing

Welcome to EIS

Please enter patient id to search:

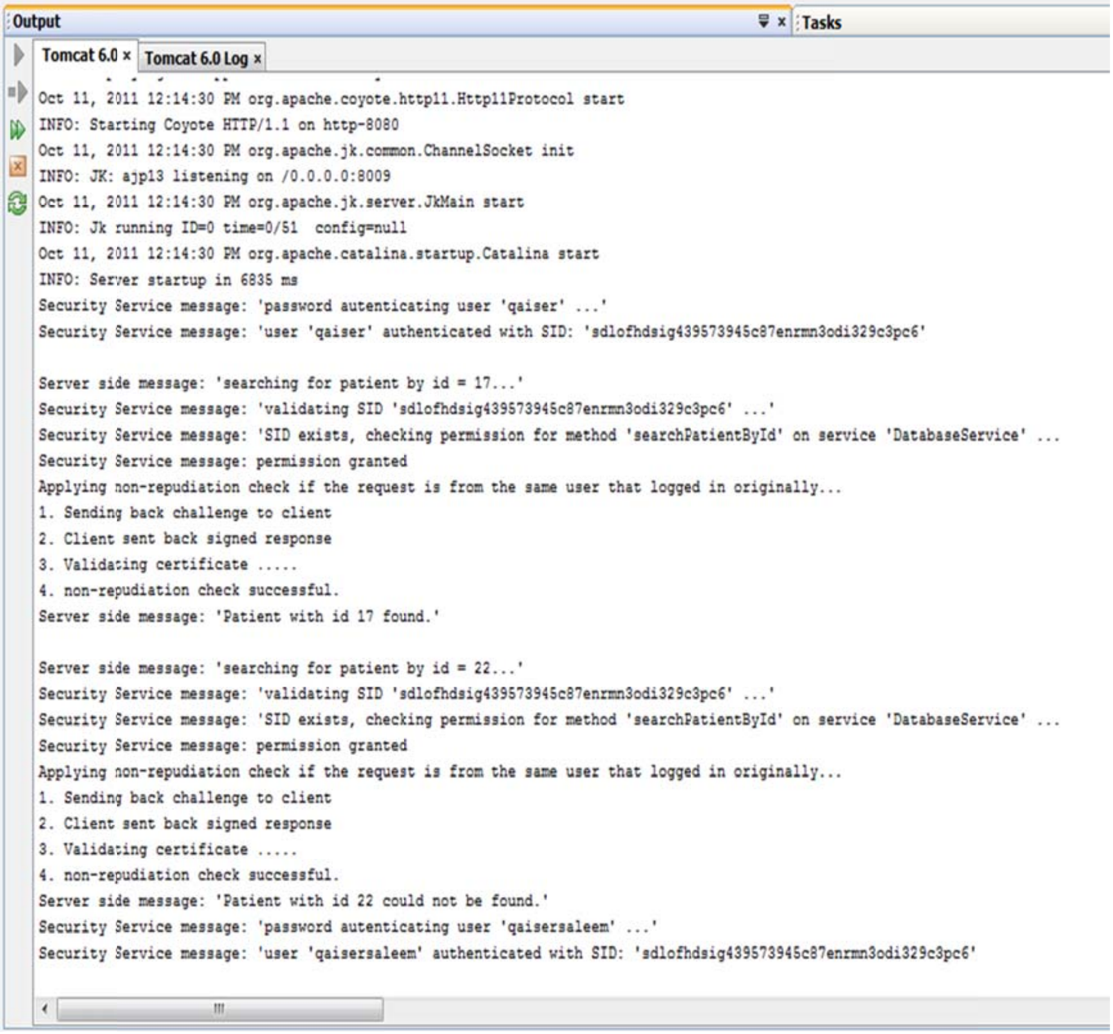
Patient Id: 22

Submit

Patient with id '22' could not be found in the database.

Figure 6.8 Patient Record is not Found

Figure 6.9 illustrates the sample output at the server side during that period of time when the client application is accessing the EIS composite application. As can be seen, security checks, authentication, authorization and non-repudiation are applied whenever a client wants to access the EIS composite application.



```
Oct 11, 2011 12:14:30 PM org.apache.coyote.http11.Http11Protocol start
INFO: Starting Coyote HTTP/1.1 on http-8080
Oct 11, 2011 12:14:30 PM org.apache.jk.common.ChannelSocket init
INFO: JK: ajp13 listening on /0.0.0.0:8009
Oct 11, 2011 12:14:30 PM org.apache.jk.server.JkMain start
INFO: Jk running ID=0 time=0/51  config=null
Oct 11, 2011 12:14:30 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 6835 ms
Security Service message: 'password autenticating user 'qaiser' ...'
Security Service message: 'user 'qaiser' authenticated with SID: 'sdlofhdsig439573945c87enrmn3odi329c3pc6'

Server side message: 'searching for patient by id = 17...'
Security Service message: 'validating SID 'sdlofhdsig439573945c87enrmn3odi329c3pc6' ...'
Security Service message: 'SID exists, checking permission for method 'searchPatientById' on service 'DatabaseService' ...
Security Service message: permission granted
Applying non-repudiation check if the request is from the same user that logged in originally...
1. Sending back challenge to client
2. Client sent back signed response
3. Validating certificate .....
4. non-repudiation check successful.
Server side message: 'Patient with id 17 found.'

Server side message: 'searching for patient by id = 22...'
Security Service message: 'validating SID 'sdlofhdsig439573945c87enrmn3odi329c3pc6' ...'
Security Service message: 'SID exists, checking permission for method 'searchPatientById' on service 'DatabaseService' ...
Security Service message: permission granted
Applying non-repudiation check if the request is from the same user that logged in originally...
1. Sending back challenge to client
2. Client sent back signed response
3. Validating certificate .....
4. non-repudiation check successful.
Server side message: 'Patient with id 22 could not be found.'
Security Service message: 'password autenticating user 'qaisersaleem' ...'
Security Service message: 'user 'qaisersaleem' authenticated with SID: 'sdlofhdsig439573945c87enrmn3odi329c3pc6'
```

Figure 6.9: Sample output at the Server side of the EIS Composite Application

6.7 Chapter Summary

This chapter presents the whole process of secure composite application development, deployment and its usage. It starts with the description of UML deployment diagram. Afterwards it illustrates the EIS BPEL workflow diagram, JBI service assembly unit and composite application deployment. At the end there is a detailed description about the scenario when a client application wants to access the composite application and how security checks are ensured.

CHAPTER 7
ANALYSIS AND RESULTS

7.0 Chapter Overview

This chapter presents the comparative study of the proposed work for qualitative analysis and results. In this chapter, two comparative studies are presented. First comparative study is presented about the "UML-SOA-Sec" and the research works very close to it. Second comparative study is presented about the "Saleem's MDS services composition framework" and other MDS services composition frameworks.

7.1 Comparative Study of the Proposed DSL "UML-SOA-Sec" with Related Work

A comparative study has been conducted for "UML-SOA-Sec" in which proposed DSL is compared with the other DSLs based on the success criteria defined by the Roy Gronmo [23] by considering the numbers of factors such as: simplicity and readability of business process model, ease of use by business process expert, use of icons to represent the security objectives and sufficient numbers of security objectives for SOA applications. Table 7.1 depicts its details and a thorough discussion is provided below.

7.1.1 Simplicity of the Business Process Model

In this section, discussions are provided regarding the approaches followed for security annotation in the proposed work and in the related work to find out that the model created using a particular approach is either simple or complex.

A business process model should be simple; it should not be messy with lots of technical details. If it contains lots of technical details then the model would be complex for a common business process expert.

Table 7.1: Comparison of “UML-SOA-Sec” with Related Work Based on Success Criteria (Continue)

S/No	Researchers	Work	Comparison of the Proposed DSL with Different Researcher’s DSLs				
			Simplicity Of Business Process Model	**Readability of Business Process Model**	**Ease of Use by Business Process Expert**	**Use Of Icons for Security Objective**	**Sufficient Number of Security Objectives for SOA Applications**
1	Muhammad Qaiser Saleem (Proposed Work)	A DSL called UML-SOA-Sec. Focusing Confidentiality, Integrity, Availability, Auditing, Non-repudiation	Yes	Easy	Easy	Yes	Yes
2	Michal Hafner and Ruth Breu [28, 38]	Security Policies regarding Confidentiality, Integrity and Availability	Little bit complex (they are preparing total of five models, two models under workflow and three under interface model)	Little bit difficult	Little bit harder for a business process expert	No	Almost Yes (However they are using three broad categories of generic Security Objectives)
3	M. Memon [139]	Security Pattern for Authentication and Non-repudiation	Little bit complex (they are preparing five models, two models under workflow and three under interface model)	Little bit difficult	Little bit harder for a business process expert	No	No

Table 7.1: Comparison of "UML-SOA-Sec" with Related Work Based on Success Criteria

S/No	Researchers	Work	Comparison of the Proposed Work by Different Researchers				
			Simplicity Of Business Process Model	**Readability of Business Process Model**	**Ease of Use by Business Process Expert**	**Use Of Icons for Security Objective**	**Sufficient Number of Security Objectives for SOA Applications**
4	Alfonso Rodríguez et al. [4, 56]	A DSL focusing different security objectives Non-repudiation, Attack Harm detection, Integrity, Privacy Access Control.	Yes	Easy	Easy	Yes	Almost Yes (However they did not mentioned the SOA architecture)
5	Christian Wolter et al. [8]	Security Policies for Confidentiality, Integrity, Authentication, Authorization, Availability and Audit	Little bit complex (they are presenting a business process model in term of three layers; business process, organization and integration)	Little bit difficult	Little Bit harder for a business process expert	Yes	Almost Yes
6	Michal Menzel et al. [20]	Security Policies for Authentication, Authorization, Trust, Data Integrity and Data Confidentiality, System Integrity and System Availability	Yes	Little bit difficult	Easy	Yes	Almost Yes

In the proposed work a single business process model is developed and security is annotated within the model with the help of proposed DSL "UML-SOA-Sec". In the proposed approach, business process expert just has to annotate the UML Activity diagram with security stereotypes with in the business process model. Security objectives are represented in the business process model as stereotypes. In this way only one diagram is created. It does not make the business process model such a mess; instead, it keeps the model simple so a business process expert can easily understand and work with it.

Unlike the proposed work, in which only one diagram is created to represent the security annotated business process model, the approach presented by Michal Hafner et al. [28, 38, 136] proposes total of five diagrams to represent the security annotated business process model. They have described two Model Views naming Workflow View and Interface View. In Workflow View, two models are created called Global Workflow Model and Local Workflow Model. While in Interface View; three models are created known as Interface Model, Role Model and Document Model. It means a total of five models are created. M. Memon [29, 139] also used the same approach in his work. It seems to be very comprehensive but complex as well. It is difficult for a common business process expert to understand and use it; who is basically not an IT expert. Furthermore, if someone spend too much time in documentations and especially focusing only one non-functional requirement i.e. "*Security*", then it would be very difficult to cope with the other functional and non- functional requirements. Hence, a security annotated business process model created using their approaches is little bit complex for a common business process expert to understand and work with it.

Similar to the proposed work, Rodriguez A. et al. is also constructing only one diagram to represent the security annotated business process model. They have proposed a DSL and used the same DSL for extending both the popular modeling languages i.e. BPMN [4] as well as the UML [56]. In their approach, a business process expert just has to annotate the business process model with security stereotypes defined in their DSL. Hence, the business process model constructed using their approach is simple.

In accordance with the proposed work, Christian Wolter et al. [8], in their approach is also constructing a single business process model and annotate it with security requirement. However, in their work the business process model is constructed in terms of three layers business process layer, organization layer and integration layer. Although, they are constructing a single business process model and annotate it with security; however, in their approach security has to be defined in terms of three layers i.e. business process layer, organization layer and integration layer. Hence, a security annotated business process model created using their approaches is little bit complex for a common business process expert to work with it.

Similar to the proposed work, Michael Menzel et al. [20], in their approach is also creating a single business process model and security is annotated with their proposed DSL SecureSOA. In their approach, a business process expert just has to annotate the business process model with security stereotypes defined in their DSL. Hence, the business process model constructed using their approach is simple

7.1.2 Readability of the Business Process Model:

In this section, discussions are provided regarding the approaches followed for security annotation in the proposed work and in the related work to find out that the model created using a particular approach is either easily readable or difficult to read.

A business process model should be easily readable. It should not be messy with lots of technical details. If it contains lots of technical details then it will affect its readability for a common business process expert.

In the proposed work a single business process model is developed and security is annotated within the model with the help of proposed DSL “UML-SOA-Sec”. In the proposed approach, business process expert just has to annotate the UML Activity diagram with security stereotypes with in the business process model. Security objectives are represented in the business process model as stereotypes. Hence, the business process model developed using the proposed approach is easily readable.

Unlike the proposed work, in which only one diagram is created to represent the security annotated business process model, the approach presented by Michal Hafner et al. [28, 38, 136] proposes total of five diagrams to represent the security annotated business process model. They have described two Model Views naming Workflow View and Interface View. In Workflow View, two models are created called Global

Workflow Model and Local Workflow Model. While in Interface View; three models are created known as Interface Model, Role Model and Document Model. It means a total of five models are created. M. Memon [29, 139] also used the same approach in his work. It seems to be very comprehensive but complex as well. It is difficult for a common business process expert to understand and use it; who is basically not an IT expert. Hence, a security annotated business process model created using their approaches is little bit difficult in terms of readability.

Similar to the proposed work, Rodriguez A. et al. is also constructing only one diagram to represent the security annotated business process model. They have proposed a DSL and used the same DSL for extending both the popular modeling languages i.e. BPMN [4] as well as the UML [56]. In their approach, a business process expert just has to annotate the business process model with security stereotypes defined in their DSL. Hence, the business process model constructed using their approach is easily readable for a common business process expert.

In accordance with the proposed work, Christian Wolter et al. [8], in their approach is also constructing a single business process model and annotate it with security requirement. However, in their approach, the business process model is constructed in terms of three layers business process layer, organization layer and integration layer. These layers address few aspects of the business process model; however, it makes the model little bit difficult in terms of readability.

In accordance with the proposed work, Michael Menzel et al. [20], in their approach is also creating a single business process model and security is annotated with their proposed DSL SecureSOA. However, their focus is security policies which contain more technical details and make the business process model little bit difficult in terms of readability.

7.1.3 Ease of Use by Business Process Expert

In this section, discussions are provided regarding the approaches followed for security annotation in the proposed work and in the related work to find out that a common business process expert can easily create a security annotated business process model using a particular approach or not.

The business process model should be easily understandable for a common business process expert so that he/she should easily work with it. A business process expert is not a security expert. Although he/she is familiar with common security notions, it is not reasonable to expect too much security knowledge from him/her to build a security policy or a security pattern.

In the proposed work, a single business process model is developed and security is annotated within the model with the help of proposed DSL "UML-SOA-Sec". In the proposed approach, business process expert just has to annotate the UML Activity diagram with security stereotypes with in the business process model. It does not require the deep understanding of how the security objectives would be realized through which security policy or security pattern. Hence, a common business process expert can easily use the proposed DSL for security annotation.

Unlike the proposed work, in which only one diagram is created to represent the security annotated business process model, the approach presented by Michal Hafner et al. [28, 38, 136] proposes total of five diagrams to represent the security annotated business process model. They have described two Model Views naming Workflow View and Interface View. In Workflow View, two models are created called Global Workflow Model and Local Workflow Model. While in Interface View; three models are created known as Interface Model, Role Model and Document Model. It seems to be very comprehensive but complex as well. It is difficult for a common business process expert to understand and use it; who is basically not an IT expert. M. Memon [29, 139] also used the same DSL in his work. Furthermore, it also required a business process expert to have a strong knowledge of security patterns as well. Hence, it is also harder for a common business process expert to work with it as it required a business process expert to create five models to represent the security annotated business process model.

Similar to the proposed work, Rodriguez A. et al. is also constructing only one diagram to represent the security annotated business process model. They have proposed a DSL and used the same DSL for extending both the popular modeling languages i.e. BPMN [4] as well as the UML [56]. In their approach, a business process expert just has to annotate the business process model with security stereotypes defined in their DSL. Hence, the business process model constructed using their approach is easily useable for a common business process expert.

In accordance with the proposed work, Christian Wolter et al. [8] is also preparing a single business process model and annotate it with security requirement. However, in their approach, the business process model is constructed in terms of three layers business process layer, organization layer and integration layer. These layers address few aspects of the business process model; however, it makes the model little bit difficult to use for a common business process expert. Furthermore, it also required the knowledge of security policies to work with it. Hence it is difficult for a common business process expert to work with their approach.

Similar to the proposed work, Michael Menzel et al. [20], in their approach is also creating a single business process model and security is annotated with their proposed DSL SecureSOA. In their approach, a business process expert just has to annotate the business process model with security stereotypes defined in their DSL. Hence, the business process model constructed using their approach is easily useable for a common business process expert.

7.1.4 Use of Icons to Represent the Security Objectives

In this section, discussions are provided to represent that either icons are used or not to represent the security objectives in the proposed work and the in related work.

Two kinds of approaches are adopted by the researchers to represent the security objectives, icons (graphical notation) and textual description. Few researchers are just using text to represent the security objectives. Although in this way a business process model can be annotated with security objectives; however, graphical representation of security objectives i.e. icons, facilitate the business process expert to incorporate the security objectives in the business process model in an easier way.

Meaningful icons are provided in the proposed DSL "UML-SOA-Sec" to represents the security objectives which facilitate a common business process expert to add security in the business process model. These icons are available at design time to incorporate security objectives in the business process models.

Similar to the proposed work, Rodriguez A. et al. [4, 56], Christian Wolter et al. [8] and Michael Menzel et al. [20] are using icons to represents the security objectives. While, Michal Hafner et al. [28, 38, 136] and M. Memon [29, 139] are not using icons to represents the security objectives, they are just using textual description to represents the security objectives.

7.1.5 Sufficient Number of Security Objectives for SOA Applications

In this section, discussions are provided regarding the number of security objectives present in in the proposed DSL and the in related work are sufficient for SOA environment or not.

SOA applications are basically distributed applications which required securing both data as well as service. Number of security objectives present in a DSL is very important because it represents that how much a DSL satisfy the security requirements of an SOA environment.

In the proposed work, after a thorough literature review (detailed discussion is provided in sections 2.9 and 2.10 of chapter 2), security objectives of SOA applications have been identified and among them the most essential five security objectives which are necessary for modelling along the business process modelling are picked. The five security objectives are Confidentiality, Integrity, Availability, Auditing and Non-repudiation. Afterwards two security mechanisms, Authentication and Authorization are described through which these security objectives would be realized. The security objectives present in the proposed DSL are sufficient for a SOA application.

In this regards, the proposed work is close to Rodriguez Alfonso et al. in which they extended the BPMN [4] and the UML [56], except that they did not mentioned the target architecture, in the case of the proposed work it is SOA. In accordance with the proposed work, Michal Hafner et al. [28, 38, 136] are dealing with three security objectives namely Confidentiality, Integrity and Availability. These security objectives are sufficient for SOA environment as authors kept them at very abstract level. Similarly the DSLs presented by the Christian Wolter et al. [8] and Michael Menzel et al. [20] are also containing almost all the required security objectives for SOA environment. Whereas, M. Memon [139] is dealing with only two security objectives, i.e. Authentication and Non-repudiation and defined patterns for them. These two security objectives are not sufficient for securing the SOA environment.

7.2 Comparative Study of the Proposed Framework with Related Frameworks

In this comparative study, proposed framework "*Saleem's MDS Services composition framework*" is compared with the other web services composition framework by considering the numbers of factors such as: phases/steps of the framework, modeling language used, focused system aspect and incorporating security modeling or not respectively. Table 7.2 depicts its details and thorough discussion is provided below.

7.2.1 Phases/Steps of Frameworks

In this section, description is provided regarding the name of the frameworks and different steps/phases present in these frameworks which are presented by researchers for services composition. Working of the framework is clear from the names of the steps/phases; however, discussions are provided to further elaborate them and perform comparison.

The proposed framework is known as "Saleem's MDS Services Composition Framework". This framework described four steps for service composition naming: UML Modelling of Service Composition, Transforming of WSDL of discovered Web Services into UML Class Diagram, Refining UML Activity Diagram of Composite Web Service, and Transforming of UML Models into WSDL and BPEL.

Table 7.2: Comparison of Saleem's MDS Services Composition Framework with Other Frameworks

S/ No	**Researchers**	**Characteristics of the MDS Web Services Composition Frameworks**				
		Title	**Phases/Steps**	**Modelling Language**	**Focused System Aspect**	**Incorporated Security Modelling**
1	Proposed by Muhammad Qaiser Saleem	Saleem's MDS Services Composition Framework	Four Steps Naming UML Modelling of Service Composition, Transforming of WSDL of discovered Web Services into UML Class Diagram, Refining UML Activity Diagram of Composite Web Service, and Transforming of UML Models into WSDL and BPEL	UML	Business Process Modelling	Incorporate Security at Step 1 and Step 3 during Business Process Modelling
2	Bart Orri¨ens et al. [50]	Service Composition Life Cycle	Four phases naming: Definition, Scheduling, Construction and Execution	UML	Business Process	No
3	Roy Grønmo and Ida Solheim [23]	Actions to build Composite Web Services	Four Phases naming: Discover Web Services, Model Composite Web Service, Implement Composite Web Service in a Workflow Engine, Publish Composite Web Service	UML Class Diagram, Activity Diagram	Business Process	No
4	Christophe Dumez et al. [85]	Composite Web Service Development Process	Five Phases naming: Import WSDLs of Web Services, UML Class Diagram generation, Generation of the interface for composite Web Service, Composite Web Service Method Definition, Composite Web Service Code Generation	UML Class Diagram, UML Activity Diagram	Business Process, Business-to-Business (B2B)	No
5	David Skogan et al. [24]	A Method	Four Steps Naming: UML Modelling and Searching of Web Services in the repository, Transformation of WSDL into UML Models for composite Web Service, Transformation of UML Models into XML document and into some execution engine, Publish the Composite Web service	UML Activity Diagram	Business Process, Modelling	No
6	Andre R. R. et al. [47]	A Methodology named Sec-MoSC (Security for Model-oriented Service Composition)	They defined thirteen steps for service composition development and grouped them in three abstraction levels, Business, Design, and Execution	BPMN	Business Process Modelling	Incorporate Security at Business process model which will propagate during the remaining steps

These are the four essential steps for web services composition starting from modeling of the composite web service to its deployment. These steps are followed in the given sequence for secure web services composition development. The security is modeled at step-1 and at step-3 along the business process modeling and propagates through the remaining steps of services composition development. The UML is used for modeling the service composition and proposed DSL "UML-SOA-Sec" is used for security modeling.

The second framework illustrated in the Table 7.2 is known as "*Service Composition Life Cycle*" and is presented by the Bart Orri¨ens et al. [50]. This framework described four phases for service composition naming: *Definition, Scheduling, Construction and Execution*. The main idea behind their approach of services composition development is to start with an abstract definition of the service composition and from this abstract definition generates the executable composed service. They have described the four steps for services composition at very abstract level. In *definition* phase a composite web services is defined along with all its requirements. In *scheduling* phase it is defined that how and when the services should be run and prepare them for execution. In *construction* phase a composite web service is constructed from the available services. Finally during the *execution* phase this constructed composed service is prepared for execution. In this approach, the UML is used for modelling the services composition; it enables the development of technology independent composition definitions, which can subsequently be mapped to a specific services composition standard e.g. the BPEL. In comparison with the proposed framework, this framework just broadly described the phases of services composition at very abstract level. Furthermore, security is not defined along the business process modeling for services composition.

The third framework illustrated in the Table 7.2 is known as "*Actions to build Composite Web Services*" and is presented by the Roy Grønmo and Ida Solheim [23]. This framework described four actions for service composition naming: Discover Web Services, Model Composite Web Service, Implement Composite Web Service in a Workflow Engine, Publish Composite Web Service. These are the four actions which would be performed in the given sequence while composite web service development. They emphasized, for the services composition modelling, one should perform two kinds of modelling; service modelling and workflow modelling. Service modelling identifies services to be exposed with their interfaces and operations (UML class diagram); while, the workflow modelling identifies the control and data flows from one service to the next service (UML activity diagram). The focus of their work is workflow modelling of the composite Web service using the UML Activity diagram. Proposed framework is also working is in the same way i.e. for service modelling, the UML class diagram is used and for workflow modelling the UML activity diagram is used. Proposed framework is close to this work; however, in the proposed framework, steps/phases are organized in more suitable way. Furthermore, they are also not defining security along the business process modeling for services composition.

The fourth framework illustrated in the Table 7.2 is known as "*Composite Web Service Development Process*" and is presented by the Christophe Dumez et al. [85]. This framework described five phases for service composition naming: Import WSDLs of Web Services, UML Class Diagram generation, Generation of the interface for composite Web Service, Composite Web Service Method Definition, and Composite Web Service Code Generation. These are the five phases which constitute the composite web services development process and these phases should be performed in the given sequence. They have defined the static aspects of the composition i.e. the interface of the services composition by the UML-Class diagram (WSDL interface and data types involved), and used the UML-activity diagram to model the dynamic aspects (the composition scenario itself, i.e. the interaction among the existing services). Proposed framework, is also working along the same direction, i.e. for services composition modelling a UML class diagram is used and for composition scenario modelling a UML activity diagram is used. However, in the proposed framework, steps/phases are organized in more suitable way. Furthermore, they are also not defining security along the business process modeling for services composition.

The fifth framework illustrated in the Table 7.2 is known as "*A Method*" and is presented by the David Skogan et al. [24]. This framework described four steps for service composition naming: UML Modelling and Searching of Web Services in the repository, Transformation of WSDL into UML Models for composite Web Service, Transformation of UML Models into XML document and into some execution engine, Publish the Composite Web service. In their framework, they have provided a way to model the coordination and the sequence of the interactions among Web services. However, in their approach, methods, input/output and data transformation are modelled as notes (i.e. comments) on the side of the workflow, which can get quite confusing when the composition flow gets complex. Furthermore, they are also not defining security along the business process modeling for services composition.

The sixth and the last framework illustrated in the Table 7.2 is known as "Methodology" for "*Sec-MoSC (Security for Model-oriented Service Composition)*" is presented by Andre R. R. et al. [47]. In this methodology, a total of thirteen steps are defined for service composition development. These thirteen steps are grouped in three abstraction levels: Business-level, Design-level and Execution-level. They have used the BPMN as a modelling language and the BPEL for services composition. Unlike the proposed framework, in this methodology a business process model is enriched with security by adding three thing; NF-Attributes, NF-Statements and NF-Actions. Furthermore, in this methodology, security requirements are identified which are represented in different views corresponding to three abstraction levels. Moreover, they have also presented general guidelines for the corresponding implementation methods against the security requirements.

7.2.2 Modeling Language Used

The modeling language used in all the frameworks is either UML or BPMN which are industry standards for business process modelling [4]. It makes no difference in these frameworks based on the selection of modeling language as they are using industry standards.

7.2.3 Focused System Aspect

The system aspect focused in all of the frameworks is "*Business Process Modeling*". It makes no difference in these frameworks based on the focused system aspect as all of these frameworks are focusing the same system aspect.

7.2.4 Incorporated Security Modeling

In the proposed Saleem's MDS Services composition framework security is incorporated during the business process modeling of services composition at two different stages i.e. at step-1 and Step-3. Unlike the proposed framework, the frameworks presented by the Bart Orri¨ens et al. [50], Roy Grønmo and Ida Solheim [23], Christophe Dumez et al. [85] and David Skogan et al. [24] did not provide any information about security i.e. they are not dealing with the aspect of "*security*" while developing a composed service. Only the framework presented by Andre R. R. et al. [47] incorporate security during the business process modeling of services composition.

7.2.5 Discussion

The main idea of this dissertation is the modeling of security objectives along the business process modeling of SOA system. In Saleem's MDS services composition framework, the main contribution is incorporating security objectives along the business process modelling of the composite application. The business process modeling is performed using the UML Activity diagram and the proposed DSL "UML-SOA-Sec" is used for modeling the security objectives. The business process modeling and security modeling is performed by a common business process expert. In the proposed framework, security is defined at two different stages. Firstly at step-1, when the overall modelling of the services composition is performed. This is the concept building stage about the services composition i.e. what functionality this composed service has to perform, which services are required to accomplish this functionality and what are the security requirements of the composed service. Secondly, at step-3, when all required services are either discovered or developed; now all the required services are available and security will be refined/redefined for modelling the services composition. This security annotated business process model will be transformed into a services composition

The main limitation of the frameworks presented in Table 7.2 is that four of them are not dealing with security at all, which is the main idea of this dissertation (Bart Orri¨ens et al. [50], Roy Grønmo and Ida Solheim [23], Christophe Dumez et al. [85] and David Skogan et al. [24]).

Only the framework presented by Andre R. R. et al. [47] have incorporated security along the services composition and presented a methodology called the "Sec-MoSC" (Security for Model-oriented Services Composition). In their methodology, total of thirteen steps are performed in three different levels, namely the Business-level, Design-level and Execution-level. They have identified the security requirements which are represented in different views corresponding to these three abstraction levels. They have also presented general guidelines for the corresponding implementation methods against the security requirements. Unlike the proposed framework, in this methodology, a business process model is enriched with security by adding three different kind of security information naming: NF-Attributes, NF-Statements and NF-Actions. Furthermore, in this methodology, security requirements are identified which are represented in different views corresponding to three abstraction levels. The beauty of a model (diagram) is its simplicity, if too many details e.g. NF-Attributes, NF-Statements and NF-Actions; are added for just one non-functional attribute "security", then the whole model will become too much messy. Moreover, unlike the proposed framework, in their methodology, a very lengthy process is presented for services composition consist of thirteen steps. Furthermore, they proposed that these steps are performed by a business process expert aided by a security expert that knows the meaning of NF-Attributes, NF-Statements and NF-Action. Unlike the proposed framework, where only one role i.e. a business process expert is required, their methodology requires an additional role of "*security expert*" to model the security requirement in the business process model.

To conclude the discussion; it can be stated that proposed Saleem's MDS services composition framework is a better choice for secure services composition. In the proposed framework security objectives are modeled during the business process modeling of composite application i.e. it define security at the early stages of software development. Furthermore it consist necessary four steps for services composition which make it simple.

7.3 Chapter Summary

This chapter presented the comparative study of the proposed work for qualitative analysis and results. In this chapter, two comparative studies are presented. Firstly, comparative study is presented about the "UML-SOA-Sec" and the research works very close to it. Secondly, comparative study is presented about the "Saleem's MDS services composition framework" and other MDS services composition frameworks.

CHAPTER 8
EVALUATION AND DISCUSSIONS

8.0 Chapter Overview

This chapter presents the evaluation of the proposed work and discussions. At the start it describes the quantitative results which are presented as evaluation of the proposed work. Afterwards, it presents the discussions about, how the research questions of the dissertation are addressed. At the end of the chapter, discussions are presented after combining both kinds of results i.e. quantitative and qualitative regarding the significance of the approach used in the proposed DSL "UML-SOA-Sec" over the approaches used in other DSLs to annotate the security in a business process diagram.

8.1 Evaluation of the Proposed Work for Quantitative Results

The main idea of the dissertation is the modeling of security objectives along the business process modeling of SOA application. In this connection a DSL named "UML-SOA-Sec" is developed. For evaluation purpose, a quantitative method (survey) has been adopted for collecting and analyzing the data. The detailed discussion about the survey method used during this work is provided in section 3.3.2 of the chapter 3. The objective of the survey method is to evaluate the proposed DSL "UML-SOA-Sec" with the previously proposed DSLs [4, 8, 28, 30, 36, 38, 56, 139] on the basis of approaches used in these DSLs for security annotation to find out the respondents response upon improvement in annotating the business process diagram with security using "UML-SOA-Sec".

Following sub-sections represents the analysis and results of the data obtained from the survey, according the dimensions of the success criteria defined Roy Gronmo [23].

8.1.1 Simplicity of the Business Process Model

When several diagrams are created to represents the business process model then it will affect its simplicity.

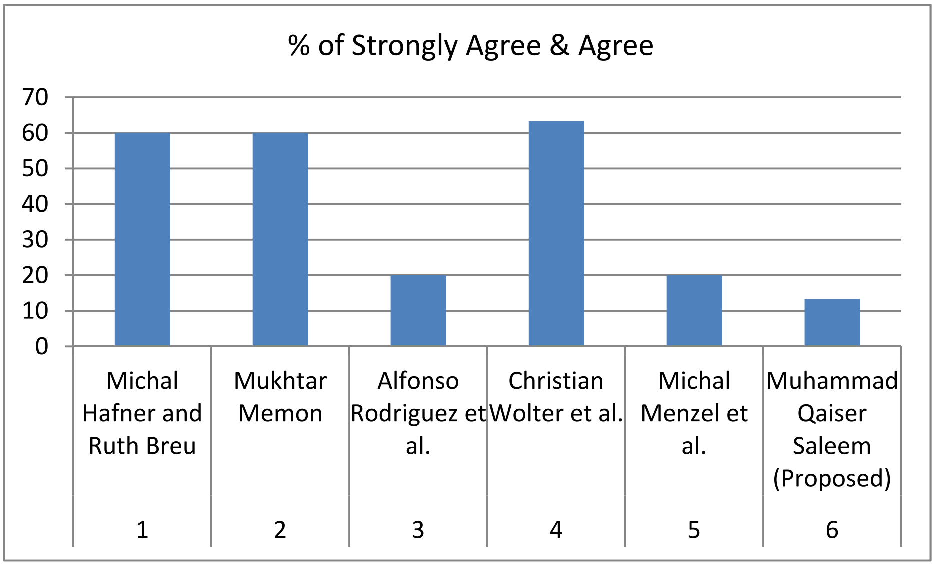
The question to measure this dimension is "*The model is messy as it contains many diagrams*". Purpose of this question is to get the feedback from the respondents about their feeling regarding the messiness of a business process model when it contains several diagrams. Table 8.1 shows the response of the respondents against this question.

Table 8.1: Response of Respondents against First Question

S/No	**The model is messy as it contains many diagrams**	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	Michal Hafner and Ruth Breu [28, 38]	8	10	7	4	1
2	Mukhtiar Memon [139]	7	11	5	6	1
3	Alfonso Rodriguez et al. [4, 56]	1	5	4	14	6
4	Christian Wolter et al. [8]	5	14	5	5	1
5	Michal Menzel et al. [20]	1	5	5	15	4
6	**Muhammad Qaiser Saleem (Proposed)**	1	3	4	15	7

In this question a negative aspects of the diagram is asked where respondents have to answer the messiness of the model when it contains several diagrams. For evaluation purposes, the data is collected from the Table 8.1 where respondents responded agree and strongly agree against this question. Based on this data, percentage is computed and column graph is simulated as shown in Graph 8.1.

Graph 8.1 Model is Messy as it contains Many Diagrams



In the proposed work a single business process model is developed and security is annotated with the proposed DSL UML-SOA-Sec. In the proposed approach, business process expert just has to annotate the UML Activity diagram with security stereotypes which is a simple approach. Unlike the proposed work, to represent the security annotated business process model, total of five diagrams are created using the approach presented by Michal Hafner et al. [28, 38, 136]. The same DSL is used by the M. Memon [29, 139] in his work. Their approach is very comprehensive; however, complex as well. Similar to the proposed work, Rodriguez A. et al. [4, 56] are constructing only one diagram to represent the business process model and uses their DSL to annotate it with security which keep the model simple. In accordance with the proposed work, Christian Wolter et al. [8] is preparing a single business process model and annotate it with security requirement; however, in their work the business process model is constructed in terms of three layers business process layer, organization layer and integration layer. These layers address few aspects of the business process model; however, it makes the model complex. Similar to the proposed work, Michael Menzel et al. [20] is also creating a single business process model and security is annotated with their proposed DSL and their approach is also simple.

The Graph 8.1 depicts that the security annotation in the business process model using "UML-SOA-Sec" is less messy as only 13 percent respondents are agreed about its messiness. According to the feedback of respondents, proposed work is better than the technique of Michal Hafner, Ruth Breu and Mukhtiar Memon in security annotation (13 percent to 60 percent). Proposed work is also better than the Alfonso Rodriguez et al. work (13 percent to 20 percent), Christian Wolter et al. (13 percent to 63 percent) and also better than Michal Menzel et al. (13 percent to 20 percent).

Hence, according to the feedback of the respondents, the security annotation in the business process model using "UML-SOA-Sec" is simpler as compared to the rest of the DSLs.

8.1.2 Readability of the Business Process Model

When a diagram has too many technical details then it will affect its readability.

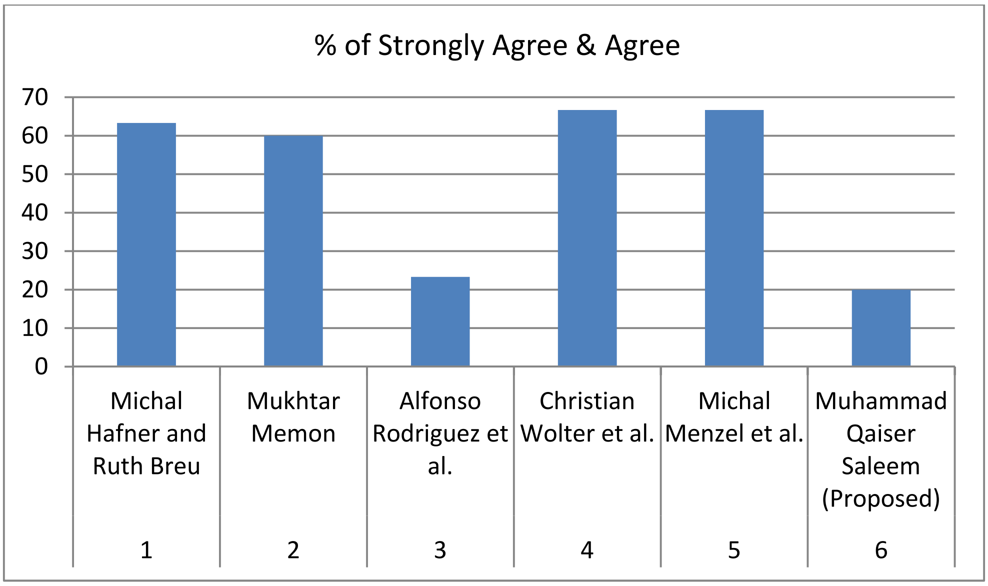
The question to measure this dimension is: "*The model has abundant Technical details*". Purpose of this question is to get the feedback from the respondents about their feeling regarding the technical details presents in a particular DSL for annotating the security in a business process diagram. Table 8.2 shows the response of the respondents against this question.

Table 8.2: Response of Respondents against Second Question

S/No	**The model has abundant Technical details**	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	Michal Hafner and Ruth Breu [28, 38]	7	12	5	5	1
2	Mukhtiar Memon [139]	6	12	5	6	1
3	Alfonso Rodriguez et al. [4, 56]	2	5	3	14	6
4	Christian Wolter et al. [8, 30]	6	14	3	4	3
5	Michal Menzel et al. [20]	5	15	3	6	1
6	**Muhammad Qaiser Saleem (Proposed)**	2	4	2	14	8

In this question a negative aspects of the diagram is asked where respondents have to answer the abundance of technical details in a model. For evaluation purposes, the data is collected from the Table 8.2 where respondents responded agree and strongly agree against this question. Based on this data, percentage is computed and column graph is simulated as shown in Graph 8.2.

Graph 8.2 Model has abundant Technical Details



In the proposed work a single business process model is developed and security is annotated with the proposed DSL UML-SOA-Sec. In the proposed approach, business process expert just has to annotate the UML Activity diagram with security stereotypes which keep the business process model simple and easily readable. Unlike the proposed work, to represent the security annotated business process model, total of five diagrams are created using the approach presented by Michal Hafner et al. [28, 38, 136]. The same DSL is used by the M. Memon [29, 139] in his work. Their approach is very comprehensive; however, it affects the readability as a model contains too much technical details. Similar to the proposed work, Rodriguez A. et al. [4, 56] are constructing only one model to represent the business process model and uses their DSL to annotate it with security which keep the model simple and easily readable. In accordance with the proposed work, Christian Wolter et al. [8] is preparing a single business process model and annotate it with security requirement; however, in their work the business process model is constructed in terms of three layers business process layer, organization layer and integration layer. These layers address few aspects of the business process model; however, it makes the model complex and difficult to read. In accordance with the proposed work, Michael Menzel et al. [20] is also creating a single business process model and security is annotated with their proposed DSL; however, their focus is security policies which contains more technical details and make the business process model little bit hard to read.

The Graph 8.2 depicts that the security annotation in the business process model using "UML-SOA-Sec" contains less technical details as only 20 percent respondents are agreed about its abundance of technical details. According to the feedback of respondents, proposed work contains less technical details than the model of Michal Hafner and Ruth Breu (20 percent to 63 percent), better than Mukhtiar Memon (20 percent to 60 percent), better than Alfonso Rodriguez et al. (20 percent to 23 percent), and also better than Christian Wolter et al. (20 percent to 67 percent) and Michal Menzel et al (20 percent to 67 percent).

Hence, according to the feedback of the respondents, the security annotation in the business process model using "UML-SOA-Sec" is more readable compared to the rest of the DSLs.

8.1.3 Ease of Use by the Business Process Expert.

A business process model with less technical details and with less number of diagrams would be easy for a business process expert to work with it.

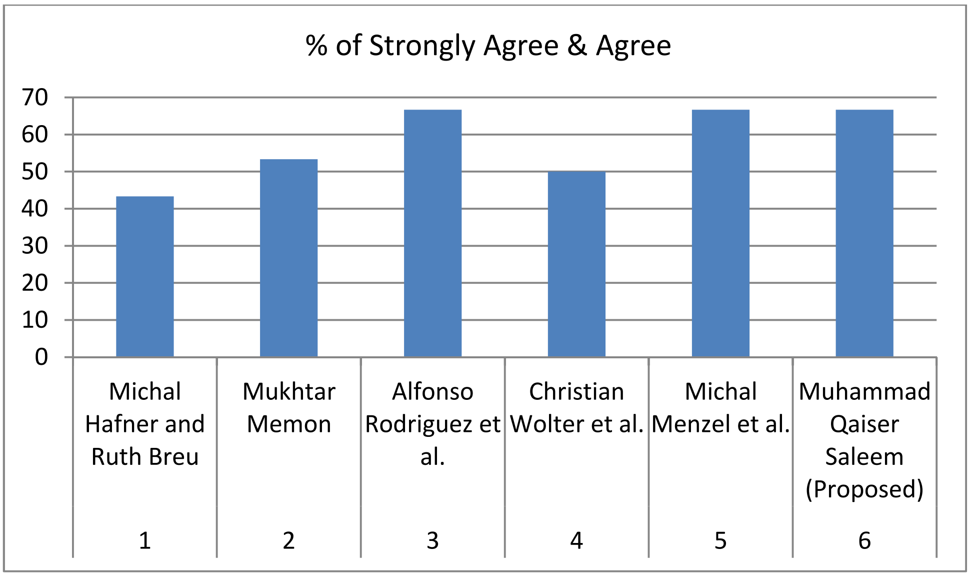
The question to measure this dimension is: “*I can easily add security objectives in the model*”. Purpose of this question is to get the feedback from the respondents that how easily he/she can annotate a business process model with security using a particular DSL. Table 8.3 shows the response of the respondents against this question.

Table 8.3: Response of Respondents against Third Question

S/No	**I can easily add security objectives in the model**	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	Michal Hafner and Ruth Breu [28, 38]	3	10	5	9	3
2	Mukhtiar Memon [139]	4	12	4	8	2
3	Alfonso Rodriguez et al. [4, 56]	6	14	4	5	1
4	Christian Wolter et al. [8]	4	11	2	7	6
5	Michal Menzel et al. [20]	5	15	4	5	1
6	**Muhammad Qaiser Saleem (Proposed)**	6	14	4	3	2

In this question, the respondents have to answer regarding their feeling about the easiness of adding the security objectives in the business process model using a particular DSL. For evaluation purposes, the data is collected from the Table 8.3 where respondents responded agree and strongly agree against this question. Based on this data percentage is computed and column graph is plotted as shown in Graph 8.3.

Graph 8.3 Security Objectives Can Easily Modeled in a Diagram



In the proposed DSL, UML-SOA-Sec, a common business process expert just has to add security objectives in the business process model as a security stereotype at a very abstract level. It does not require the deep understanding of how the security objectives would be realized through which security policy or security pattern. Unlike the proposed work, Michal Hafner et al. [28, 38, 136] and M. Memon [29, 139] are defining security policies and security patterns for the security objectives. Their approach is little bit difficult for a common business process expert as it required strong knowledge of security policies and patterns. Similar to the proposed work, a model constructed using the DSL proposed by Rodriguez A. et al. [4, 56] is easy for a business process expert because they just have to annotate the model with security stereotypes. Unlike the proposed work, Christian Wolter et al. [8] are defining security policies for security objectives that's why model construction is a little bit difficult for a common business process expert as it requires knowledge of security policies. Similar to the proposed work, Michael Menzel et al. [20] provide the modeling enhancement to add security in a business process model which can easily be used by a common business process expert.

As can be seen from Graph 8.3, it is easier to add security objectives in the business process model using the "UML-SOA-Sec", as 67 percent respondents are agreed with it. According to the feedback of respondents, proposed approach is better than the approach of Michal Hafner and Ruth Breu in annotating the business process model with security (67 percent to 43 percent), better than Mukhtiar Memon (67 percent to 53 percent), equals to Alfonso Rodriguez et al. (67 percent to 67 percent), better than Christian Wolter et al. (67 percent to 50 percent) and again equals to Michal Menzel et al. (67 percent to 67 percent)

Hence, it's easier for a business process expert to annotate the business process diagram using "UML-SOA-Sec".

8.1.4 Use of Icons in a DSL to Represent the Security Objective

Two kinds of approaches are adopted by the researchers to represent the security objectives, icons (graphical notation) and textual description. Few researchers are just using text to represent the security objectives. Although in this way a business process model can be annotated with security objectives; however, graphical representation of security objectives i.e. icons, facilitate the business process expert to incorporate the security objectives in the business process model in an easier way.

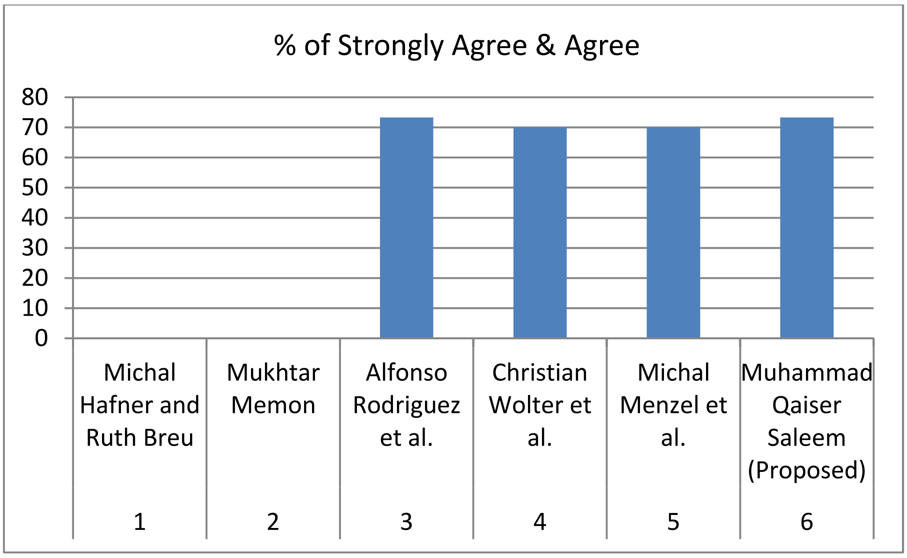
The question to measure this dimension is: "*The security icons make it easier for me to add security objectives in the model*". Purpose of this question is to get the feedback from the respondents regarding how easily they can annotate a business process model using security icons present in the DSLs. Table 8.4 shows the response of the respondents against this question.

Table 8.4 Response of Respondents against Forth Question

S/No	**The security icons make it easier for me to add security objectives in the model**	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	Michal Hafner and Ruth Breu [28, 38]	0	0	0	0	0
2	Mukhtiar Memon [139]	0	0	0	0	0
3	Alfonso Rodriguez et al. [4, 56]	8	14	3	3	2
4	Christian Wolter et al. [8]	6	15	1	7	1
5	Michal Menzel et al. [20]	6	15	4	4	1
6	**Muhammad Qaiser Saleem (Proposed)**	8	14	4	3	1

In this question, the respondents have to answer regarding their feeling about the easiness of adding the security objectives in the business process model using icons present in the DSLs. For evaluation purposes, the data is collected from the Table 8.4 where respondents responded agree and strongly agree against this question. Based on this data percentage is computed and column graph is plotted as shown in Graph 8.4.

Graph 8.4 Security Icons are Provided for Security Modeling



The proposed DSL, UML-SOA-Sec is using meaningful icons to represents the security objectives. These icons are available at design time of business process modeling to incorporate security objectives in the business process models. Similar to the proposed work, Rodriguez A. et al. [4, 56], Christian Wolter et al. [8] and Michael Menzel et al. [20] are using icons to represents the security objectives. While, Michal Hafner et al. [28, 38, 136] and M. Memon [29, 139] are not using icons to represents the security objectives, rather they are just using textual description to represents the security objectives.

As can be seen from Graph 8.4, the percentage of response from the respondents is almost same (73 percent, 70 percent, 70 percent, 73 percent) except where there are no security icons are provided in the DSLs by the researchers i.e. Michal Hafner, Ruth Breu and Mukhtiar Memon.

It is easy to add the security objectives in a business process diagrams when the security icons are provided in the DSLs. Hence, usage of icons (graphical notation) in "UML-SOA-Sec" to represent the security objectives makes it easy for a business process expert to add security in a business process model.

8.1.5 Sufficient Number of Security Objectives Present in the DSLs.

Number of security objectives present in a DSL is very important because it represents that how much a DSL satisfy the security requirements of an SOA environment.

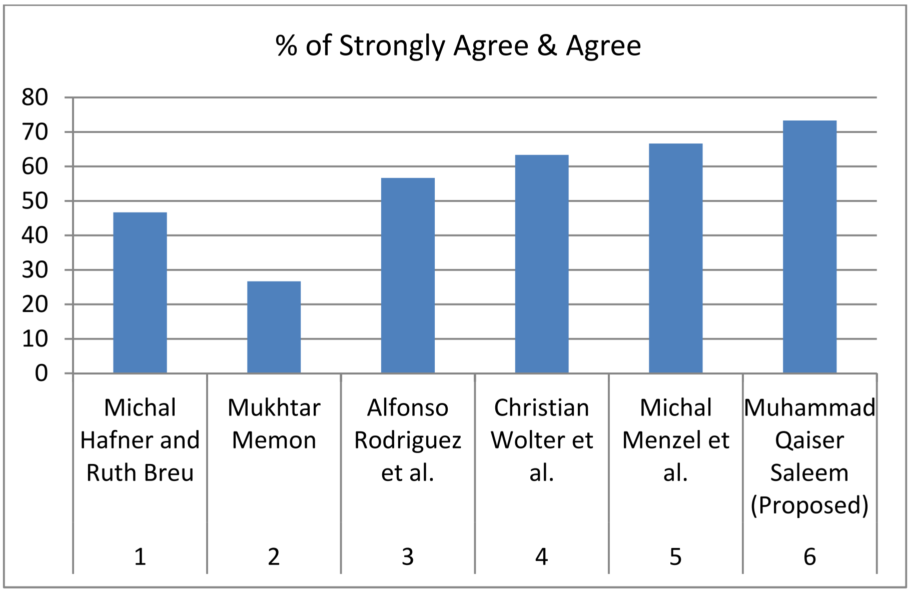
The question to measure this dimension is: "*The numbers of security objectives present in the security DSL are sufficient for SOA environment*". Purpose of this question is to get the feedback from the respondents about their satisfaction regarding the number of security objectives present in a particular DSL. Table 8.5 shows the response of the respondents against this question.

In this question, the respondents have to answer that the numbers of security objectives present in the different security DSLs are sufficient for SOA environment. For evaluation purposes, the data is collected from the Table 8.5 where respondents responded agree and strongly agree against this question. Based on this data, percentage is computed and a column graph is plotted as shown in Graph 8.5.

Table 8.5: Response of Respondents against Fifth Question

S/No	The number of security objectives present in the security DSL are sufficient for SOA environment	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	Michal Hafner and Ruth Breu [28, 38]	3	11	2	9	5
2	Mukhtiar Memon [139]	2	6	2	14	6
3	Alfonso Rodriguez et al. [4, 56]	4	13	2	9	2
4	Christian Wolter et al. [8]	5	14	4	7	0
5	Michal Menzel et al. [20]	6	14	3	5	2
6	**Muhammad Qaiser Saleem (Proposed)**	7	15	4	3	1

Graph 8.5 No of Security Objectives Present in DSLs are Sufficient for SOA Environment



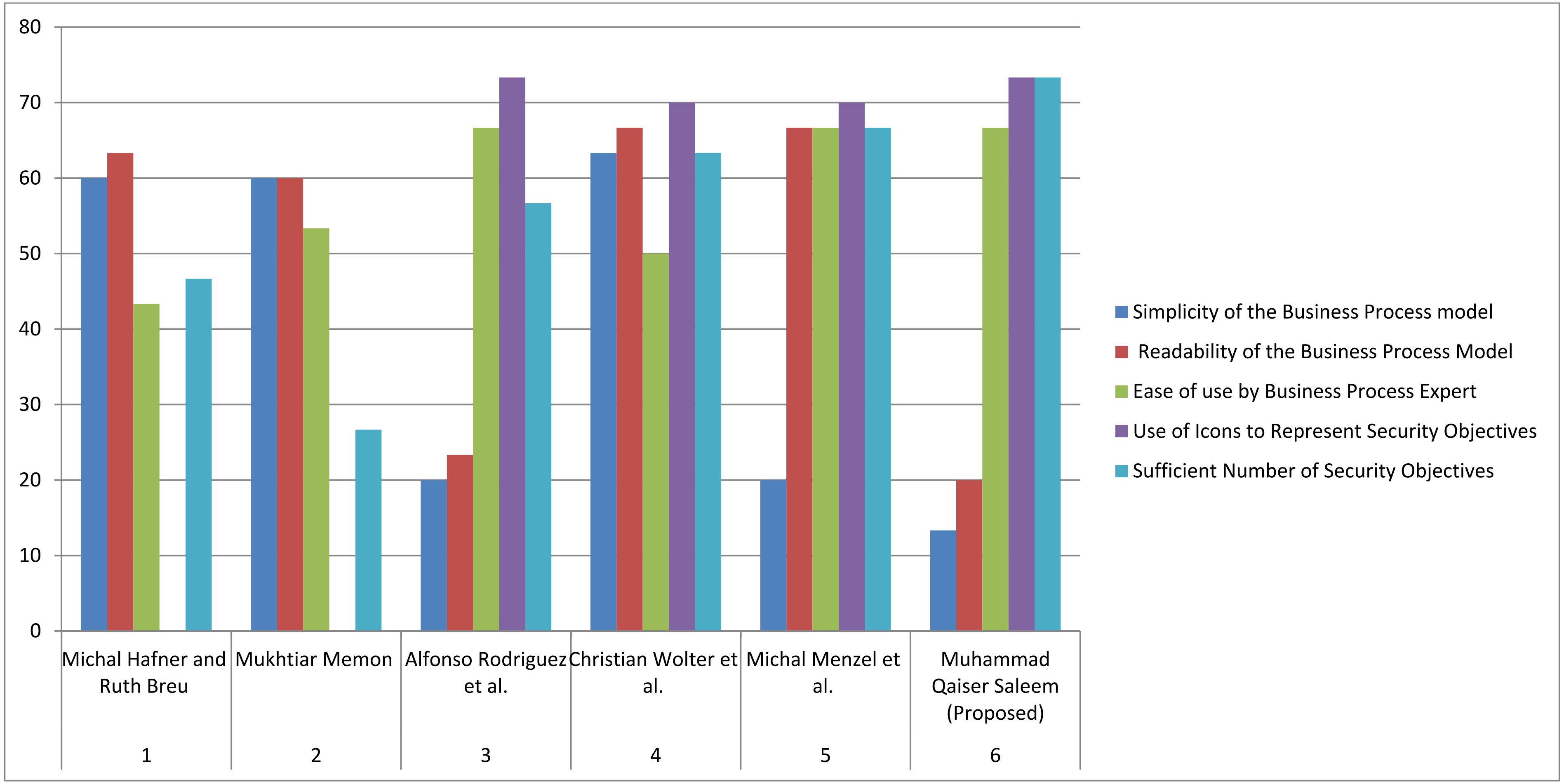
In the proposed work, five security objectives are identified which are essential to be modelled within a business process model for an SOA application. Afterwards two security mechanisms are described through which these security objectives would be realized. Different DSLs proposed by different researchers contains different number of security objectives. In this regards, the proposed work is close to Rodriguez Alfonso et al. in which they extended the BPMN [4] and the UML [56], except that they did not mentioned the target architecture, in the case of the proposed work it is SOA. In accordance with the proposed work, Michal Hafner et al. [28, 38, 136] are dealing with three security objectives namely Confidentiality, Integrity and Availability. These security objectives are sufficient for SOA environment as authors keep them at very abstract level. Similarly the DSLs presented by the Christian Wolter et al. [8] and Michael Menzel et al. [20] are also containing almost all the required security objectives for SOA environment. Whereas, M. Memon [139] is dealing with only two security objectives, i.e. Authentication and Non-repudiation and defined patterns for them. These two security objectives are not sufficient for securing the SOA environment.

As can be seen from Graph 8.5, in the proposed approach, sufficient numbers of security objectives regarding SOA environment are provided as 73% respondents are agreed with it. According to the feedback of respondents, proposed approach is better than the model of Michal Hafner and Ruth Breu in containing sufficient number of security objectives for SOA environment (73 percent to 47 percent), better than Mukhtiar Memon (73 percent to 27 percent), better than Alfonso Rodriguez et al. (73 percent to 57 percent), better than Christian Wolter et al. (73 percent to 63 percent). and better than Michal Menzel et al (73 percent to 67 percent).

Hence, “UML-SOA-Sec” contains sufficient numbers of security objectives for SOA environment.

The whole discussion of evaluation UML-SOA-Sec is summarized and presented in the Graph 8.6.

Graph 8.6 Evaluation of “UML-SOA-Sec” with Other DSLs based on the Success Criteria



8.2 Discussions

This section, initially presents the discussions related to addressing the research questions of the dissertation. Afterwards, discussions are presented, after combining both kinds of results i.e. qualitative (comparative study of proposed work, discussed in chapter 7) and quantitative (evaluation of proposed work, discussed in section 8.1 of this chapter) regarding the significance of the approach used in the proposed DSL "UML-SOA-Sec" over the approaches used in other DSLs to annotate the security in a business process diagram.

8.2.1 Addressing the Research Questions

The main idea of the dissertation is the modeling of security objectives along the business process modeling of SOA application. SOA security scenarios are investigated and it is found that security is not incorporated during the early development stages of an SOA application i.e. during the business process modeling because of two main reasons. Firstly, there is no clear identification of security objectives to be modelled during the business process modelling of the SOA application. Secondly, current general purpose modelling languages, like the UML, lack the modelling of QoS attributes; security is among the most important QoS attributes. Having these reasons in mind, following two research questions are developed and addressed during this dissertation.

- *Research Question 1: What are the essential security objectives to be modeled during the Business Process modeling for SOA applications?*

One of the reasons of not incorporating security at early stages of SOA application development is that there is not clear identification of security objectives to be modelled during the business process modelling of SOA applications. From the thorough literature review (detailed discussions are provided in sections 2.9 and 2.10 of chapter 2), security objectives of SOA applications have been identified and among them the most essential security objectives which are necessary for modelling along the business process modelling were picked. The selected security objectives are Confidentiality, Integrity, Availability, Auditing and Non-repudiation. These are the essential five security objectives which fulfills the requirements of SOA applications and which would be modeled during the business process modeling of SOA applications. Furthermore, two security mechanisms are also identified through which these security objectives would be realized naming Authentication and Authorization.

Hence, with the identification of the security objectives to be modeled during the business process modeling of SOA applications, the first research question is successfully addressed.

- *Research Question 2: How can the general purpose modelling language be enriched to specify security objectives in a Business Process Model of an SOA application in a formalized manner?*

Another reason of not incorporating security at early stages of SOA application development is that the current general purpose modelling languages, like the UML, lack the modelling of QoS attributes; security is among the most important QoS attribute. There should be some formal means through which security would be modelled in a business process model for secure SOA application development. Having this very essential aspect in mind a DSL naming "UML-SOA-Sec" is developed.

There are several ways of defining the DSL (detailed discussions are provided in section 2.6.2 of chapter 2); however, to gain the benefits of the general purpose modelling language, DSLs are defined in terms of general purpose modelling languages like the UML. General purpose modelling languages can easily be customized by the extension mechanism provided by the language itself and the DSL can be defined according to the domain of interest by extending the general purpose modelling language. In case of the UML the extension mechanism is known as the UML Profile (detailed discussions are provided in sections 2.6.5 and 2.6.6 of chapter 2). Tools are available for the general purpose modelling languages which support the definition and usage of the DSL.

In case of proposed DSL "UML-SOA-Sec", a metamodel is defined to represent the abstract syntax, illustrating the security objectives and the security mechanisms identified for the SOA environment. Afterwards, UML profiling mechanism is used to represent the concrete syntax in which security objectives are defined as stereotypes in the UML. After the definition of the domain specific UML-profile, a general-purpose modelling tool can easily be specialized and these domain specific stereotypes are made available at the modelling level in the form of annotation (detailed discussions are provided in section 4.1 of chapter 4).

In the proposed approach, the business process expert, only have to model the security objectives along the business process modelling of the SOA application. Later on, the architectural team will implement the security mechanisms based on the security objectives present in the business process model. In this way, during the implementation, the architectural team gets the idea what security objective the business process expert wants, and they have the flexibility to implement a potentially better security solution.

Hence, with the development of the proposed DSL "UML-SOA-Sec", the second research question is successfully addressed.

Furthermore, SOA applications are basically composition of services. These services may be scattered across the Internet. To accomplish a business activity, Web services are composed using services composition languages/standards e.g. the BPEL, which define the execution order of services invocations and their interaction patterns. However these services composition languages/standards do not deal with the early stages of software development. Several Web services composition frameworks/methods are proposed for Web services composition; where emphasis is also given to the early phases of services composition and the whole life cycle is defined. These frameworks described the different combinations of steps/phases of services composition; however, the notion of security is neglected in almost all of them i.e. security objectives are not defined during the business process modeling of SOA applications developed through these frameworks. Having these reasons in mind, third research question is developed and addressed during this dissertation.

- *Research Question3: What are the essential steps/phases for a services composition framework? And at which steps/phases of the services composition framework, security objectives would be defined/modelled?*

In this dissertation a framework named "*Saleem's MDS Service composition framework*" has been developed for the model-driven development of secure Web services composition. Its details are provided in section 4.2 of chapter 4. In this framework, the four most important steps are selected and organized among the steps discussed by different researchers. The four steps of the proposed framework are: UML Modelling of Service Composition, Transforming of WSDL of discovered Web Services into UML Class Diagram, Refining UML Activity Diagram of Composite Web Service, and Transforming of UML Models into WSDL and BPEL. These are the four essential steps for web services composition, starting from modeling of the composite web service to its deployment.

However, the main contribution in the proposed framework is incorporating the security objectives along the business process modelling. The business process modelling is performed using the UML Activity diagram and the proposed DSL "UML-SOA-Sec" is used for the security modelling. The proposed framework facilitates the business process expert in modelling the security along the business process modelling for the service composition. In the proposed framework, security is defined at two different stages; firstly, at step-1, when the overall modelling of the service composition is performed. This is the concept building stage about the service composition i.e. what functionality this composed service has to perform, which services are required to accomplish this functionality and what are the security requirements for this composed service. Secondly, at step-3, when all required services are either discovered or developed; at this stage, all the required services are available and security will be refined/redefined during the modelling of service composition. This security annotated business process model will be transformed into a service composition.

Hence, with the development of the proposed framework "*Saleem's MDS Service composition framework*", the third research question is successfully addressed.

8.2.2 Significance of Proposed DSL "UML-SOA-Sec"

In the following sub-sections, discussions are presented after combining both kinds of results i.e. qualitative (comparative study of proposed work, discussed in chapter 7) and quantitative (evaluation of proposed work, discussed in section 8.1 of this chapter), regarding the significance of the approach used in the proposed DSL “UML-SOA-Sec” over the approaches used in other DSLs to annotate the security in a business process diagram.

8.2.2.1 Simplicity of the Business Process Model

A business process model should be simple; it should not be messy with lots of technical details. If several diagrams are created to represents the business process model then it will also affect its simplicity. If it contains lots of technical details then the model would be complex for a common business process expert.

In the proposed work a single business process model is developed and security is annotated within the model with the help of proposed DSL “UML-SOA-Sec”. In the proposed approach, business process expert just has to annotate the UML Activity diagram with security stereotypes with in the business process model. Security objectives are represented in the business process model as stereotypes. In this way only one diagram is created to represent the security annotated business process model. It does not make the business process model such a mess; instead, it keeps the model simple so a business process expert can easily understand and work with it as described in Table 7.1. The statement of simplicity of the business process model using the proposed approach is also supported by the feedback of the respondents. The feedback shows that 87 percent of respondents have agreed on the simplicity of the business process model using the proposed approach which is quite high as compared to the other approaches as described in Graph 8.1.

In summary, the security annotation in the business process model using “UML-SOA-Sec” is simpler as compared to the rest of the DSLs.

8.2.2.2 Readability of the Business Process Model

When a diagram has too many technical details then it will affect its readability. The business process model shall be easy to understandable for a common business process expert so that he/she should easily see what is going on.

The business process model developed using the proposed approach is easily readable as it just annotates the UML Activity diagram with security stereotypes presented in “UML-SOA-Sec”. In the proposed approach, only one diagram is created to represent the security annotated business process model. It does not make the business process model such a mess; instead, it keeps the model simple so a business process expert can easily understand and work with it as described in Table 7.1. The statement of easy readability of the business process model using the proposed approach is also supported by the feedback of the respondents. Security annotation in the business process model using “UML-SOA-Sec” contains less technical details which is evident from the feedback showing 80 percent respondents who have agreed that the model is easily readable. Which is quite high as compared to the other approaches as described in Graph 8.2.

Hence, the security annotation in the business process model using “UML-SOA-Sec” is easily readable compared to the rest of the DSLs.

8.2.2.3 Ease of use by Business Process Expert

A business process expert is not a security expert. Although he/she is familiar with common security notions, it is not reasonable to expect too much security knowledge from him/her to build a security policy or a security pattern. A business process model with less technical details and with less number of diagrams would be easy for a business process expert to work with it.

In the proposed approach, a common business process expert just has to add security objectives in the business process model as a security stereotypes. It does not require the deep understanding of how the security objectives would be realized through which security policy or security pattern. Furthermore, in the proposed approach, only one diagram is created to represent the security annotated business process model. Hence, it is easier for a common business process expert to add security objectives in the business process model using the “UML-SOA-Sec” as described in Table 7.1. Feedback of the respondents also supports the statement as 67 percent respondents are agreed with it, which is quite high as compared to the other approaches as described in Graph 8.3.

In summary, the security annotation in the business process model using "UML-SOA-Sec" is easily useable for a common business process expert compared to the other DSLs.

8.2.2.4 Use of Icons to Represent Security Objectives

Meaningful graphical representation of security objectives i.e. icons, facilitate the business process expert to incorporate the security objectives in a business process model in an easier way.

By providing icons to represents the security objectives, DSLs facilitate a common business process expert to add security objectives in a business process model. Meaningful icons (graphical notations) are provided in the proposed DSL "UML-SOA-Sec" to represents the security objectives which facilitate a common business process expert to add security objectives in the business process model as described in Table 7.1. In relation to this, the feedback of the respondents also supports the claim that the meaningful icons make it easy for common business process expert to add security objectives in the business process model. As the response is almost same against all the DSLs except where the DSLs did not provide icons as can be seen in the Graph 8.4.

Hence, the usage of icons to represent the security objectives in the proposed DSL "UML-SOA-Sec" makes it easy for a common business process expert to add security in a business process model.

8.2.2.5 Sufficient Number of Security Objectives for SOA Applications

SOA applications are basically distributed applications requiring securing both data as well as services. Number of security objectives present in a DSL is very important because it represents that how much a DSL satisfy the security requirements of an SOA environment.

In the proposed work, five security objectives are identified which are essential to be modelled within a business process model for an SOA application naming: Confidentiality, Integrity, Availability, Auditing and Non-repudiation. Afterwards security mechanisms (Authentication and Authorization) are described through which these security objectives would be realized. The security objectives presented in the proposed DSL are sufficient for a SOA application as described in Table 7.1. The feedback of the respondents also supports the claim that proposed DSL contains the sufficient numbers of security objectives for SOA environment as 73% respondents are agreed with it which is quite high as compared to the other approaches as described in Graph 8.5.

Hence, the proposed DSL “UML-SOA-sec” contains the sufficient number security objectives to be modeled for the business process modeling of SOA applications.

In conclusion; it can be stated that the business process diagram created using the proposed DSL “UML-SOA-Sec” is simple, easily readable and easily useable by a common business process expert. Moreover, the use of “icons” to represent the security objectives in “UML-SOA-Sec” makes it easy for a common business process expert to add security in a business process diagram. Furthermore, the “UML-SOA-Sec” contains the sufficient number of security objectives for an SOA environment.

8.3 Chapter Summary

This chapter presented the evaluation of the proposed work and discussions. At the start it described the quantitative results which are presented as evaluation of the proposed work. Afterwards, it presented the discussions about, how the research questions of the dissertation are addressed. At the end of the chapter, discussions are presented after combining both kinds of results i.e. quantitative and qualitative regarding the significance of the approach used in the proposed DSL “UML-SOA-Sec” over the approaches used in other DSLs to annotate the security in a business process diagram.

CHAPTER 9
CONCLUSIONS AND FUTURE WORKS

9.0 Chapter Overview

This chapter describes the conclusions of the research work in the area of security modelling along the business process modelling for web services based SOA applications. It starts with the research summary followed by the addressing the research problems, achievements of objectives and contributions. Lastly, the chapter discusses recommendations/suggestions for future work.

9.1 Research Summary

This research work was committed to investigate the challenges in security modelling for the development of secure SOA systems, which are developed in a vulnerable Internet environment. The investigation covered the security problems faced by the current SOA system development practices and the realization was one of the reasons for the security problems is that security objectives are not incorporated during the early phases of software development i.e. during modelling of SOA applications. This might happen due to many reasons i.e. unclear security objectives for the SOA environment, lack of security modelling techniques in current general purpose modelling languages, current software development practices where security is considered as an afterthought and implemented in an ad-hoc manner, left on to the developer etc.

Incorporating security objectives during the early stages of software development improve the "*security*" of SOA applications. In this connection, a Domain Specific Language called "UML-SOA-Sec" was proposed; it contains two things, security objectives and security mechanisms. The security objectives are the essential security objectives to be modelled in a business process model for SOA applications; whereas, security mechanisms are those mechanisms through which these security objectives would be realized. For the DSL definition, the UML language was relied on, which is an industry standard for the business process modelling. The UML-profiling mechanism was used to extend the UML and security stereotypes were defined to specify security objectives during the business process modelling. Being able to express security objectives in a widely used design notation like UML for SOA systems, helps to save time and effort during the implementation and verification of security in the system. Furthermore, specifying security objectives at the abstract level helps the architectural team in choosing different, and potentially better, security mechanisms

SOA applications are basically compositions of services. Saleem's MDS services composition framework was proposed for the secure web services composition. The most essential four phases for the services composition framework were identified and later on, those steps where security would be modelled were also identified. The "UML-SOA-Sec" was used for security modelling during this research work.

For the demonstration of this work, proposed DSL "UML-SOA-Sec" and proposed MDS Services composition framework is implemented on a real world case-study (healthcare application). It is an example of an SOA environment. Security was modelled along the business process modelling of the Entity Identification Service (EIS) using a UML activity diagram and annotating it with security using the "UML-SOA-Sec". Later on, a composite application was developed and deployed on an application server which was basically a composition of business services as well as security services. This composite application can be accessed via a client application; security checks are applied whenever some client application wants to access this composite application.

At the end, comparative analysis and evaluation of the proposed work is performed. Initially two comparative studies are conducted. In the first comparative study, comparative analyses are presented about the "UML-SOA-Sec" and the research works very close to it. In the second comparative study, comparative analyses are presented about the "Saleem's MDS services composition framework" and other MDS services composition frameworks. Findings are represented in the form of discussion as well as tables. Finally evaluation of the proposed DSL "UML-SOA-Sec" is performed using the survey method. In this survey, the proposed DSL is compared with the previously proposed DSLs [4, 8, 28, 30, 36, 38, 56, 139] on the basis of approaches used in these

DSLs for security annotation to find out the respondents response upon improvement in annotating the business process diagram with security using "UML-SOA-Sec". Personally administered questionnaire are used as a survey instrument for data collection. Data is analyzed and findings are presented in the form of discussion as well in the form graphs.

9.2 Addressing the Research Problems

In chapter 1, research problems to be addressed in the dissertation in the perspectives of modelling of security objectives along the business process modeling of SOA applications and secure service composition were described. In this section, a discussion is presented that how those problems have been addressed in the proposed approach.

9.2.1 A DSL "UML-SOA-Sec" Development

An investigation has been carried out on SOA security scenarios and it was found that in the present practices of software development, security objectives are not incorporated in the early stages of SOA application development because of two main reasons. Firstly, current general purpose modelling languages, like the UML, lack the modelling of QoS attributes; security is among the most important QoS attribute. Secondly, there is not a clear identification of security objectives to be modelled during the business process modelling of SOA applications.

There should be some formal means through which security would be modelled in a business process model for secure SOA application development. Having this very essential aspect in mind the "UML-SOA-Sec" is developed. The security objectives of SOA applications have been identified and among them the most essential security objectives which are necessary for modelling along the business process modelling were picked. The selected security objectives are confidentiality, integrity, availability, auditing and non-repudiation. Two security mechanisms are also identified through which these security objectives would be realized naming authentication and authorization. A metamodel is defined, illustrating the security objectives and the security mechanisms. Afterwards, UML profiling mechanism was used for the definition of these security objectives as stereotypes in the UML. After the definition of the domain specific UML-profile, a general-purpose modelling tool can easily be specialized and these domain specific stereotypes are made available at the modelling level in the form of annotation.

A business process expert, who is not a security expert, will only model the security objectives along the business process modelling of the SOA application. Later on, an architectural team will implement the security mechanisms based on the security objectives present in the business process model. Then, the architectural team will have flexibility; they will get an idea of what the security objective business expert wants and the flexibility will enable them to implement a potentially better security solution.

9.2.2 MDS Service Composition Framework development

SOA applications are basically composition of services and these services may be scattered across the Internet. Web Services are composed using service composition languages/standards e.g. BPEL, which define the execution order of services invocation and their interaction pattern; however, these standards/languages do not deal with the early stages of the software development. Several web service composition frameworks/methods are proposed; however, notion of security is neglected in almost all of them. Security is not defined during the business process modelling of services composition. In this work, Saleem's framework for the model-driven development of a secure web services composition has been developed. In this framework, the most important four steps are selected among the steps discussed by different researchers. The four steps of the proposed framework are UML modeling of services composition, transformation of the WSDL of the discovered web services into a UML class diagram, refining the UML Activity diagram of the composite web service and transformation of UML diagrams into WSDL and BPEL. However, the main contribution is incorporating security objectives along the business process modelling. The business process modelling is performed using the UML activity diagram and the proposed DSL "UML-SOA-Sec" is used for the security modelling. The proposed framework facilitates the business process expert in modelling the security along the business process modelling for the service composition. In the proposed framework, security is defined at two different stages; firstly, at step-1, when the overall modelling of the service composition is performed. This is the concept building stage about the service composition i.e. what functionality this composed service has to perform, which services are required to accomplish this functionality and what are the security requirements for this composed service. Secondly, at step-3, when all required services are either discovered or developed; at this time, all the required services are available and security will be refined/redefined for the modelling service composition. This security annotated business process model will be transformed into a service composition.

9.3 Achievement of Objectives

The following are the objectives of this research work and their respective achievements:

- Modeling of security objectives along the business process modelling of SOA applications.

  The objective is achieved by developing a DSL named "*UML-SOA-Sec*" consist of the essential security objectives to be modeled along the business process modeling of SOA applications.

- Development of MDS services composition framework where security objectives are modelled during the business process modeling of services composition.

  The objective is achieved by developing a framework named "*Saleem's MDS services composition framework*" for the secure web services composition in which security is modeled at early stages of software development using "UML-SOA-Sec".

- Comparison and evaluation of security enhanced business process model using the proposed approach with the related approaches.

  The objective is achieved by performing two comparative studies and a survey. In first comparative study, proposed "UML-SOA-Sec" is compared with the previously proposed DSLs and in second comparative study proposed framework "Saleem's MDS services composition framework" is compared with the previously proposed frameworks. At the end a survey is performed to evaluate the proposed DSL “UML-SOA-Sec” with the previously proposed DSLs on the basis of approaches used in these DSLs for security annotation to find out the percentage improvement in annotating the business process diagram with security using “UML-SOA-Sec”

9.4 Contributions

The main contributions of the research work are as follow:

- *“UML-SOA-Sec”*. The Proposed DSL facilitates the modeling of security objectives along the business process modeling of SOA applications. It is comprised of the most essential security objectives to be modelled for the SOA applications. General purpose modelling language UML was extended by providing a metamodel and a UML profile. MagicDraw UML modelling tool is used which support the definition and usage of DSL. The proposed DSL facilitates the business process expert in modelling security objectives along the business process modelling of SOA application.

- *“Saleem’s MDS Services Composition Framework*”. The proposed framework facilitates the secure composition of the web services. It is comprised of the four most essential steps for web services composition. In this framework, security objectives are defined along the business process modelling using the proposed DSL “UML-SOA-Sec”.

9.5 Future Works

As the nature of knowledge is, each work has to have some limitations to ensure the future research continuation in this field. Therefore, some of the directions are identified in which further work can be done:

- During this work, the focus was on an SOA environment. One can explore the possible usage of the proposed DSL in other architectural environments. It may result in either the enhancement or reduction of some security stereotypes depending upon the architecture of the target environment.

- At the current stage, a DSL is proposed containing the essential security objectives to be modelled for the SOA application i.e. security is defined at the PIM. Afterwards, a developer will implement these security objectives in an application i.e. code generation from the model is not automatic. One can explore how to automate the process i.e. security objectives present in the DSL can automatically be transformed into code. This may result in the definition of the metamodel at PSM or ISM level and definition of the transformation rules between the metamodels at the PIM and PSM or ISM.

- Different approaches are used to model the security along the business process modelling and automatically generate the code from these models. A common assumption in these approaches is that the security services at the target platform realize pre-defined security patterns and security mechanisms. For example, an authorization service implements the Role-based Access Control (RBAC) pattern. As a result, an authorization service, which realizes the RBAC pattern, is not capable of implementing the Attribute Based Access Control (ABAC) or Context-based Access Control patterns. The same is the case for the other security services like authentication and non-repudiation etc. This renders these approaches inflexible to realizing patterns in current SOA security scenarios, where security services have to realize various patterns, depending upon the attributes of the service requester and its security domain. Therefore, how these MDS approaches are made capable to generate security configurations for security services to realize required patterns is another area of study.

REFERENCES

[1] R. S. Ulrich Lang, "Top SOA Security Concerns & OpenPMF Model-Driven Security," *ObjectSecurity white-paper, Topics Cloud Computing and Security Management,* 2009.

[2] D. G. Firesmith, "Engineering Security Requirements," *Journal of Object Technology, 2(1):53-58,* 2003.

[3] D. Y. Xie, Shi Zhang, Tao Jia, Xiang-Yang Liang, Zao-Qing Yao, Jun-Feng, "An Approach for Describing SOA," in *International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2006*, 2006, pp. 1-4.

[4] F.-M. E. Rodríguez Alfonso, Piattini Mario, "A BPMN Extension for the Modeling of Security Requirements in Business Processes," *IEICE - Trans. Inf. Syst.,* vol. E90-D, pp. 745-752, 2007.

[5] M. E. Orlowska*, et al.*, "Representing Web Services with UML: A Case Study," in *Service-Oriented Computing - ICSOC 2003*. vol. 2910, ed: Springer Berlin / Heidelberg, 2003, pp. 17-27.

[6] M. A. B. Siming Kou, Amit Sangroya, "Modeling security for service oriented applications," in *ACM Proceedings of the Fourth European Conference on Software Architecture: Companion,*, Copenhagen, Denmark, 2010, pp. 294-301.

[7] A. Rodriguez, Fernandez-Medina, E., Piattini, M., "Security requirement with a UML 2.0 profile," in *The First International Conference on Availability, Reliability and Security, ARES 2006.* , 2006, p. 8 pp.

[8] M. M. Christian Wolter, Christoph Meinel, "Modelling Security Goals in Business Processes," *Proc. GI Modellierung 2008, GI LNI 127, Berlin, Germany,* vol. pp. 197 - 212, March 2008.

[9] B. M. Mohsen Asadi, Nima Kaviani, Dragan Gašević, Marko Bošković, Marek Hatala, "Model-driven development of families of Service-Oriented Architectures," presented at the Proceedings of the First International Workshop on Feature-Oriented Software Development, Denver, Colorado, 2009.

[10] M. Alam, "Model Driven Realization of Dynamic Security Requirements in Distributed Systems," *Phd Dissertation, University of Insbruck, Austria,* 2007.

[11] Microsoft, "The Future of Information Technology: Growing the Talent Critical for Innovation," *Microsoft white paper, July 2006, http://research.microsoft.com/en-us/um/redmond/events/fs2006/papers/TheFutureofInformationTechnology.pdf (Date Accesed 13-11-2011).*

[12] M. T. Yuichi Nakamura, Takeshi Imamura, and Koichi Ono, "Model-driven security based on a Web services security architecture," in *IEEE International Conference on Services Computing, 2005*, 2005, pp. 7-15 vol.1.

[13] S. Hanna and M. Munro, "Fault-Based Web Services Testing," in *Fifth International Conference on Information Technology: New Generations, 2008. ITNG 2008.* , 2008, pp. 471-476.

[14] T. Erl, "Service-Oriented Architecture: Concepts, Technology, and Design," *Prentice Hall PTR Upper Saddle River, NJ, USA ©2005,* 2005.

[15] T. Erl, "SOA Principle of Service Design," *Prentice Hall,* 2008.

[16] G. Lewis, A. Morris, E., Simanta, S., Wrage, L., "Common Misconceptions about Service-Oriented Architecture," in *Sixth International IEEE Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems, 2007. ICCBSS '07.* , 2007, pp. 123-130.

[17] P. N. Asit Dan, "Dependable Service-Oriented Computing," *IEEE Internet Computing, March/April 2009, pp. 11–15,* 2009.

[18] R. K. Philip Bianco, Paulo Merson, "Evaluation of Service-Oriented Architecture," *Software Engineering Institute/ Carnegie Mellon,* vol. Technical Report, CMU/SEI-2007-TR-015, September 2007, 2007.

[19] M. P. Papazoglou, "Service-oriented computing: concepts, characteristics and directions," in *Proceedings of the Fourth International Conference on Web Information Systems Engineering, WISE 2003.*, 2003, pp. 3-12.

[20] I. T. Michael Menzel, Christoph Meinel, "Security Requirements Specification in Service-Oriented Business Process Management," in *International*

Conference on Availability, Reliability and Security, ARES '09., 2009, pp. 41-48.

[21] A. N.-s.-m. Christophe Dumez, Jaafar Gaber, Maxime Wack, "Modeling and Specification of Web Services Composition Using UML-S," presented at the Proceedings of the 2008 4th International Conference on Next Generation Web Services Practices, 2008.

[22] Bart Orri¨ens*, et al.*, "Model Driven Service Composition," 2003.

[23] I. S. Roy Grønmo, "Towards Modeling Web Service Composition in UML," *INSTICC Press. Presented at The 2nd International Workshop on Web Services: Modeling, Architecture and Infrastructure, Porto, Portugal,* April 2004.

[24] D. Skogan*, et al.*, "Web service composition in UML," in *Eighth IEEE International Enterprise Distributed Object Computing Conference, EDOC 2004. Proceedings.* , 2004, pp. 47-57.

[25] OMG, "OMG Model Driven Architecture," *http://www.omg.org/mda/ (Date Accesed 30-10-2011).*

[26] A. Rodríguez*, et al.*, "Towards CIM to PIM Transformation: From Secure Business Processes Defined in BPMN to Use-Cases," in *Business Process Management*, ed, 2007, pp. 408-415.

[27] M. C. David Basin, Marina Egea, "A decade of model-driven security," presented at the Proceedings of the 16th ACM symposium on Access control models and technologies, Innsbruck, Austria, 2011.

[28] R. B. Michal Hafner, Berthold Agreiter, "SECTET: an extensible framework for the realization of secure inter-organizational workflows," *Emeral, Internet Research,* vol. Vol.16 No. 5 pp. pp.491-506, 2006.

[29] M. H. Mukhtiar Memom, Ruth Breu, "SECTISSIMO: A Platform-independent Framework for Security Services," *MODSEC08 Modeling Security Workshop,* 2008.

[30] M. M. Christian Wolter, Christoph Meinel, Andreas Schaad, Philip Miseldine, "Model-driven business process security requirement specification," *J. Syst. Archit.,* vol. 55, pp. 211-223, 2009.

[31] J. Jurjens, "UMLsec: Extending UML for Secure Systems Development-Tutorial," presented at the Proceedings of the 5th International Conference on The Unified Modeling Language, 2002.

[32] D. B. Torsten Lodderstedt, Jürgen Doser, "SecureUML: A UML-Based Modeling Language for Model-Driven Security," presented at the Proceedings of the 5th International Conference on The Unified Modeling Language, 2002.

[33] C. M. Michael Menzel, "A Security Meta-model for Service-Oriented Architectures," in *IEEE International Conference on Services Computing, SCC '09'.* , 2009, pp. 251-259.

[34] D. M. B. E. Kamsties, and B. Paech, "Detecting Ambiguities in Requirements Documents using Inspections," *WISE01: Proceedings of the 1st workshop on inspection in software engineering, pages 68-80,* 2001.

[35] B. B. Stephan Roser, "A categorization of collaborative business process modeling techniques," in *Seventh IEEE International Conference on E-Commerce Technology Workshops, 2005.* , 2005, pp. 43-51.

[36] I. T. Michael Menzel, Christian Wolter, Christoph Meinel, "SOA Security - Secure Cross-Organizational Service Composition," *Proc. Stuttgarter Softwaretechnik Forum (SSF), Fraunhofer IRB-Verlag, Stuttgart, Germany, pp. 41 - 53,* November 2007. .

[37] J. D. David Basin, Torsten Lodderstedt, "Model driven security: From UML models to access control infrastructures," *ACM Trans. Softw. Eng. Methodol.,* vol. 15, pp. 39-91, 2006.

[38] R. B. Michal Hafner, "Security Engineering for Service-Oriented Architectures," *Springer-Verlag Berlin Heidelberg,* 2009, .

[39] F. Satoh, *et al.*, "Adding Authentication to Model Driven Security," in *International Conference on Web Services, 2006. ICWS '06.* , 2006, pp. 585-594.

[40] M. M. Tomaž Lukman, "Model-Driven Engineering and its introduction with metamodeling tools," *9th International PhD Workshop on Systems and Control: Young Generation Viewpoint, Izola, Slovenia,* October 2008.

[41] S. Luján-Mora, Trujillo, Juan, Song, Il-Yeol, "Extending the UML for Multidimensional Modeling," in *UML 2002, LNCS 2460, pp. 290-304, 2002*, ed: Springer-Verlag Berlin Heidelberg 2002, 2002, pp. 265-276.

[42] V. L. Stefanov, Beate Korherr, Birgit, "Extending UML 2 Activity Diagrams with Business Intelligence Objects," in *Data Warehousing and Knowledge Discovery*, ed, 2005, pp. 53-63.

[43] R. D. Passerone, W. Ben Hafaiedh, I. Graf, S. Ferrari, A. Mangeruca, L. Benveniste, A. Josko, B. Peikenkamp, T. Cancila, D. Cuccuru, A. Gerard, S. Terrier, F. Sangiovanni-Vincentelli, "Metamodels in Europe: Languages, Tools, and Applications," *Copublished by the IEEE CS and the IEEE CASS,* vol. 26, pp. 38-53, 2009.

[44] J. Jürjens, "UMLsec: Extending UML for Secure Systems Development," in *«UML» 2002 — The Unified Modeling Language*, ed, 2002, pp. 1-9.

[45] M. Mernik*, et al.*, "When and how to develop domain-specific languages," *ACM Comput. Surv.,* vol. 37, pp. 316-344, 2005.

[46] J. D. Achim D. Brucker, "Metamodel-based UML Notations for Domain-specific Languages," *4th International Workshop on Language Engineering (ATEM 2007)* 2007.

[47] L. Baresi, Chi, Chi-Hung, Suzuki, Jun, Souza, Andre, Silva, Bruno, Lins, Fernando, Damasceno, Julio, Rosa, Nelson, Maciel, Paulo, Medeiros, Robson, Stephenson, Bryan, Motahari-Nezhad, Hamid, Li, Jun, Northfleet, Caio, "Incorporating Security Requirements into Service Composition: From Modelling to Execution," in *Service-Oriented Computing*. vol. 5900, ed: Springer Berlin / Heidelberg, 2009, pp. 373-388.

[48] Wikipedia, "Business Process Model and Notation," *http://en.wikipedia.org/wiki/Business_Process_Model_and_Notation (Date Accesed 03-11-2011),* 2011.

[49] J. Jurjens, "Developing Secure System with UMLsec From business process to implementation," *Computing Laboratory University of Oxford GB,* 2001.

[50] B. Y. Orriëns, Jian Papazoglou, Mike, "Model Driven Service Composition," *Service-Oriented Computing - ICSOC 2003, Springer Berlin / Heidelberg, pp-75-90,* vol. 2910, pp. 75-90, 2003.

[51] C. Dumez, "Approche dirig´ee par les mod`eles pour lasp´ecification, la v´erification formelle et la miseen oeuvre de services Web compos´es " *PhD Dissertation, Universite de Technologie Belfort-Montbeliard, France.,* 2010.

[52] ORACLE, "Web Services Security: What's Required To Secure A Service-Oriented Architecture," *An Oracle White Paper,* 2008.

[53] Y. N. Fumiko Satoh, Nirmal K. Mukhi, Michiaki Tatsubori, Kouichi Ono, "Methodology and Tools for End-to-End SOA Security Configurations," in *IEEE Congress on Services - Part I, 2008.*, 2008, pp. 307-314.

[54] W. C. Klarl Heiko, Emig Christian, "Identity Management in Business Process Modelling: A Model-Driven Approach," in *Konzepte, Technologien, Anwendungen: 9. Internationale Tagung Wirtschaftsinformatik, Wien, 25 -27. Februar 2009. Teil 1*, H. R. Hansen*, et al.*, Eds., ed Wien: Ã–sterreichische Computer Gesellschaft, 2009, pp. 161-170.

[55] M. Alam, "Model Driven Security Engineering for the Realization of Dynamic Security Requirements in Collaborative Systems," *MoDELS 2006 Workshops, LNCS 4364, pp. 278–287. Springer-Verlag Berlin Heidelberg,* 2007.

[56] A. Rodríguez, Fernández-Medina, Eduardo, Piattini, Mario, "Towards a UML 2.0 Extension for the Modeling of Security Requirements in Business Processes," in *Trust and Privacy in Digital Business*, ed, 2006, pp. 51-61.

[57] W. M. P. Van der Aalst, Dumas, M., Ter Hofstede, A. H. M., "Web service composition languages: old wine in New bottles?," in *Euromicro Conference, 2003. Proceedings.*, 2003, pp. 298-305.

[58] A. V.-M. Lidia Fuentes-Fernández, "An Introduction to UML Profiles," ***UP****GRADE, the European Journal for the Informatics Professional,* vol. V, No. 2, April 2004.

[59] S. Brahe and K. Østerbye, "Business Process Modeling: Defining Domain Specific Modeling Languages by Use of UML Profiles," in *Model Driven Architecture – Foundations and Applications*, ed, 2006, pp. 241-255.

[60] Q. F. Hassan, "Important Aspects of SOA: An Entry Point for Starters," *Annals. Computer Science Series , Mirton Publishing House, Timisoara, Vol- VII, Phase 2,,* 2009.

[61] J. Y. Chung, "Services Sciences, Management and Engineering," *IBM Report, http://www.research.ibm.com/ssme (Date Accessed 10-10-2011),* 2005.

[62] HP, "Service Oriented Architecture Services,," *http://www.hp.com/hpinfo/newsroom/press/2005/050628c.html (Date Accesed 13-11-2011),* 2005.

[63] HP, "HP's approach to Service-Oriented Architecture (SOA),," *http://h71028.www7.hp.com/enterprise/cache/329749-0-0-225-121.html (Date Accesed 13-11-2011),* 2011.

[64] SAP, "Sap Services: Enterprise SOA Services," *http://www.sap.com/index.epx#/services-and-support/index.epx (Date Accesed 13-11-2011),* 2011.

[65] SAP, "SOA Middleware," *http://www.sdn.sap.com/irj/sdn/nw-soa (Date Accesed 13-11-2011),* 2011.

[66] ORACLE, "Oracle Service-Oriented Architecture," *http://www.oracle.com/us/technologies/soa/index.html (Date Accesed 13-11-2011),* 2011.

[67] ORACLE, "Service Oriented Architecture: Whats New," *http://www.oracle.com/technetwork/topics/soa/whatsnew/index.html (Date Accesed 13-11-2011),* 2011.

[68] ORACLE, "Oracle® SOA Suite, Best Practices Guide, 10g Release 3 (10.1.3.3.0), E10971-01," *http://download.oracle.com/technology/tech/soa/soa_best_practices_1013x_drop3.pdf (Date Acceded 13-11-2011),* December 2007.

[69] CISCO, "Cisco IT Department Deploys Innovative Cisco AON Solutions.," *http://www.cisco.com/en/US/prod/collateral/modules/ps6438/product_promotion0900aecd802f5ecc.html (Date Acceseed 13-11-2011),* 2011.

[70] W. T. Tsai, "Service-oriented system engineering: a new paradigm," in *IEEE International Workshop on Service-Oriented System Engineering, SOSE 2005.* , 2005, pp. 3-6.

[71] W. T. Tsai, *et al.*, "Perspectives on Service-Oriented Computing and Service-Oriented System Engineering," in *Service-Oriented System Engineering, 2006. SOSE '06. Second IEEE International Workshop*, 2006, pp. 3-10.

[72] L. B. Liam O'Brien, Paulo Merson "Quality Attributes and Service-Oriented Architectures " *Software Engineering Institute/ Carnegie Mellon,* vol. Technical Note : CMU/SEI-2005-TN-014 *September 2005*

[73] S. G. Antonio Bucchiarone, "A Survey on Services Composition Languages and Models," *International Workshop on Web Services Modeling and Testing (WS-MaTe 2006),* 2006.

[74] Y. S. Haitao Song, Yingyu Yin, Shixiong Zheng, "Dynamic Weaving of Security Aspects in Service Composition," in *Second IEEE International Workshop on Service-Oriented System Engineering, 2006. SOSE '06.* , 2006, pp. 189-196.

[75] Antonio Bucchiarone and S. Gnesi, "A Survey on Services Composition Languages and Models," *International Workshop on Web Services Modeling and Testing (WS-MaTe 2006),* 2006.

[76] OASIS, "Universal Description, Discovery and Integration Speci-fication Version 3.0.2. OASIS UDDI Technical Committee," *http://uddi.org/pubs/uddi_v3.htm (Date Accesed 14-11-2011),* 2004.

[77] W. W. Web, "XML-Signature Syntax and Processing W3C Recommendation " *http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/ (Date Accesed 30-10-2011),* 12 February 2002

[78] D. Smyth, "XML Encription," *http://dotnetslackers.com/articles/xml/XMLEncryption.aspx (Date Accessed 30-10-2011),* 15 Dec 2006.

[79] C. W. M. M. a. C. Meinel., "Access Control for Cross-Organizational Web Service Composition.," *Journal of Information Assurance and Security, 2:155-160.,* 2007.

[80] Benatallah*, et al.*, "Service Composition: Concepts, Techniques, Tools and Trends," *Book Chapter. Service-Oriented Software Engineering: Challenges and practices, chapter 3, pages pp. 48-66. IDEA Group-2005,* 2005.

[81] A. B. Maurice Beek, Stefania Gnesi, "A Survey on Service Composition Approaches: From Industrial Standards to Formal Methods," *IEEE CS Press. In Technical Report 2006TR-15, Istituto. PP 15--20,* 2006.

[82] A. B. Maurice ter Beek, Stefania Gnesi, "Web Service Composition Approaches: From Industrial Standards to Formal Methods," in *Second International Conference on Internet and Web Applications and Services, ICIW '07.* , 2007, pp. 15-15.

[83] J. Z. Martin Henkel, "Architectures for Service-oriented Processes," *Procedings of the Nordic Conference on Web Services (NCWS'04), Växsjö, Sweden,* 2004.

[84] C. Peltz, "Web services orchestration and choreography," *Computer,* vol. 36, pp. 46-52, 2003.

[85] C. G. Dumez, J. Wack, M., "Web Services Composition using UML-S: A Case Study," in *GLOBECOM Workshops, 2008, IEEE*, 2008, pp. 1-6.

[86] R. S. R. S. Aguilar-Savén, "Business process modelling: Review and framework," *International Journal of Production Economics,* vol. 90, pp. 129-149, 2004.

[87] N. Damij, "Business process modelling using diagrammatic and tabular Techniques," *Business Process Management Journal,* vol. 13, pp. 70-90, 2007.

[88] A. R. Riad, Hassan, Q. F., "Service-Oriented Architecture – A New Alternative to Traditional Integration Methods in B2B Applications," *Journal of Convergence Information Technologies (JCIT),* vol. VoL 3, No 1, March 2008.

[89] M. J. A. Mana A., Rudolph C., Vivas J. L., "A business process-driven approach to security engineering," in *Proceedings of 14th International Workshop on Database and Expert Systems Applications, 2003.* , 2003, pp. 477-481.

[90] S. Sadiq, *et al.*, "Modeling Control Objectives for Business Process Compliance," in *Business Process Management*, ed, 2007, pp. 149-164.

[91] A. G. G. Stoneburner, and A. Feringa, "*Risk Management Guide for Information Technology Systems.*," *National Institute of Standards and Technology, special publication 800-30 edition,* July 2002.

[92] F.-M. E. Rodríguez Alfonso, Piattini Mario, "Towards CIM to PIM Transformation: From Secure Business Processes Defined in BPMN to Use-Cases," in *Business Process Management*, ed, 2007, pp. 408-415.

[93] D. Basin*, et al.*, "Model driven security: From UML models to access control infrastructures," *ACM Trans. Softw. Eng. Methodol.,* vol. 15, pp. 39-91, 2006.

[94] R. France and B. Rumpe, "Model-driven Development of Complex Software: A Research Roadmap," in *Future of Software Engineering, FOSE '07*, 2007, pp. 37-54.

[95] E. Seidewitz, "What models mean," *Software, IEEE,* vol. 20, pp. 26-32, 2003.

[96] M. B. a. V. G. Sami Beydeda, "Model-Driven Software Development," *Springer-Verlag Berlin Heidelberg,* 2005.

[97] C. Atkinson and T. Kuhne, "Model-driven development: a metamodeling foundation," *Software, IEEE,* vol. 20, pp. 36-41, 2003.

[98] OMG, "Meta Object Facility (MOF) 2.0 Core Specification," *OMG Available Specification, ptc/04-10-15,* 2005.

[99] C. B. Hyoseob Kim, "Developing Software Metrics Applicable to UML Models," *6th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE),* 2002.

[100] Wikipedia, "Metamodeling," *http://en.wikipedia.org/wiki/Metamodeling (Date Accesed 21-11-2011).*

[101] B. Selic, "A Systematic Approach to Domain-Specific Language Design Using UML," in *10th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing, ISORC '07.* , 2007, pp. 2-9.

[102] J. S. Kai Chen, Sandeep Neema, "Toward a semantic anchoring infrastructure for domain-specific modeling languages," *in 5th International Conference on Embedded Software, EMSOFT 2005, pp. 35-43,* 2005.

[103] B. Selic, "The pragmatics of model-driven development," *Software, IEEE,* vol. 20, pp. 19-25, 2003.

[104] B. M. Giovanni Giachetti, Oscar Pastor, "Integration of Domain-Specific Modelling Languages and UML Through UML Profile Extension Mechanism," *International Journal of Computer Science and Applications,* © *Technomathematics Research Foundation,* vol. 6, No. 5, pp 145-174, 2009.

[105] K. H. James Bruck, "Customizing UML: Which Technique is Right for You?," *IBM , http://www.eclipse.org/modeling/mdt/uml2/docs/articles/Customizing_UML2_Which_Technique_is_Right_For_You/article.html (Date Accesed 18-11-2008),* 2008.

[106] J. Farhad, "The UML Extension Mechanism," *Dept of Computer Science, University College London, http://www.cs.ucl.ac.uk/staff/ucacwxe/lectures/3C05-02-03/aswe15-essay.pdf (Date Accesed 18-11-2011),* 2002.

[107] C. M. Michael Menzel, "SecureSOA Modelling Security Requirements for Service-Oriented Architectures," in *IEEE International Conference on Services Computing (SCC)*, 2010, pp. 146-153.

[108] M. Q. Saleem, *et al.*, "Model Driven Security Frameworks for Addressing Security Problems of Service Oriented Architecture," *ITSim 2010,* 2010.

[109] T. Lodderstedt, "Model driven security: from UML models to access control architectures," *PhD Disertation, Albert-Ludwig University of Freiberg, Germany,* March 2004.

[110] V. A. B. Daniela Cascini Peixoto, Ana Paula Atayde, Eduardo Borges, Rodolfo Ferreira Resende, Isaiah Clarindo Padua. , "A Comparison of BPMN and UML 2.0 Activity Diagrams," *VII Brazilian Symposium on Software Quality, Florianópolis* 2008.

[111] O. M. Group, "Business Process Modeling Notation Specification.," *OMG Final Adopted Specification dtc/06-02-01,* Febrary 2006.

[112] I. Essential Strategies, "The Unified Modeling Language (UML)," *http://www.essentialstrategies.com/publications/modeling/uml.htm (Date Accessed 06-08-2012),* 1999.

[113] S. W. Ambler, "The Object Primer 3rd Edition Agile Model Driven Development with UML 2," *Cambridge University Press,* 2004.

[114] Wikipedia, "Acitivity Diagram," *http://en.wikipedia.org/wiki/Activity_diagram (Date Accessed 06-08-2012),* 30 October 2006.

[115] D. Frankel, "Model Driven Architecture: Applying MDA to Enterprise Computing," *Wiley,* 2003.

[116] M. R. Tal Garfinkel, Dan Boneh, "Flexible OS Support and Applications for Trusted Computing (2003)," *In 9th hot topics in operating systems (HOTOS-IX),* pp. 145--150, 2003.

[117] D. Gollmann., "Computer Security," *John Wiley & Sons,,* 1998.

[118] I. L. Ajay Tipnis, "Security - A major imperative for a service oriented architecture," *EDS White Paper,* June 2008.

[119] P. C. Ramarao Kanneganti, "SOA Security," *MANNING,* December 2007.

[120] C. Criteria, "Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model," *CCIMB-99-031, Version 2.1,* 1999.

[121] E. F.-B. M. Schumacher, D. Hybertson, F. Buschmann, P. Sommerlad, "*Security Patterns : Integrating Security and Systems Engineering (Wiley Software Patterns Series)*." *John Wiley & Sons,,* March 2006.

[122] N. Nagaratnam*, et al.*, "Business-driven application security: from modeling to managing secure applications," *IBM Syst. J.,* vol. 44, pp. 847-867, 2005.

[123] F. E. B. Delessy N. A., "A Pattern-Driven Security Process for SOA Applications," in *Third International Conference on Availability, Reliability and Security, ARES 08.* , 2008, pp. 416-421.

[124] K. R. Jun Han, Khan K. M., "Security-Oriented Service Composition and Evolution," in *13th Asia Pacific Software Engineering Conference, APSEC 2006.* , 2006, pp. 71-78.

[125] I. T. Michael Menzel, Christian Wolter, Christoph Meinel, "SOA Security - Secure Cross-Organizational Service Composition," *In Proc. Stuttgarter Softwaretechnik Forum (SSF), Fraunhofer IRB-Verlag, Stuttgart, Germany, pp. 41 - 53,* , November 2007.

[126] S. Johnston, "Modeling security concerns in service-oriented architectures," *IBM developerWorks,* 2004.

[127] T. Erl, "An overview of the WS Security Framework," *http://www.soaspecs.com/ws-security.php,* Date Accessed (8-March-2011).

[128] P. Tan, Jun, Han, Mueller, I., Kapuruge, M., Versteeg, S., "SOABSE: An approach to realizing business-oriented security requirements with Web Service security policies," in *IEEE International Conference on Service-Oriented Computing and Applications (SOCA), 2009*, 2009, pp. 1-10.

[129] B. G. Marianne Swanson, "Generally Accepted Principles and Practices for Securing Information Technology Systems," *National Institute of Standards and Technology Technology Administration U.S. Department of Commerce,* 1996.

[130] J.-C. L. Algirdas Aviz ̌ ienis, Brian Randell, "Fundamental Concepts of Dependability," *Research Report N01145, LAAS-CNRS,* April 2001.

[131] V. Jonnaganti, "An Integrated Security Model for the Management of SOA," *Master Thesis University of Gothenburg, Sweden,* May 2009.

[132] R. S. Poore. (1999, Generally Accepted System Security Principles. *by International Information Security Foundation*. Available: http://www.infosectoday.com/Articles/gassp.pdf

[133] C. M. Basin David, Egea Marina, Schläpfer Michael, "Automatic Generation of Smart, Security-Aware GUI Models " *Springer-Verlag Berlin Heidelberg, ESSoS 2010, LNCS* vol. 5965, pp. 201-217, 2010.

[134] J. Jürjens, "UMLsec: Extending UML for Secure Systems Development," in *«UML» 2002 — The Unified Modeling Language*. vol. 2460, J.-M. Jézéquel*, et al.*, Eds., ed: Springer-Verlag Berlin Heidelberg, 2002, pp. 1-9.

[135] J. J. Bastian Best, Bashar Nuseibeh, "Model-Based Security Engineering of Distributed Information Systems using UMLsec," *ICSE '07 Proceedings of the*

29th international conference on Software Engineering , IEEE Computer Society Washington, DC, USA ©2007 pp. 581-590, 2007.

[136] K. B. Ruth Breu, Michael Hafner, Jan, Jürjens, Guido Wimmel,Volkmar Lotz, "Key Issues of a Formally Based Process Model for Security Engineering," *In Proceedings of the 16th International Conference on Software & Systems Engineering and their Applications (ICSSEA03),* 2003.

[137] B. R. Alam M. M., Breu M., "Model driven security for Web services (MDS4WS)," in *8th International Multitopic Conference, 2004. Proceedings of INMIC 2004.* , 2004, pp. 498-505.

[138] M. H. M. Alam, M. Memon, and P. Hung, "Modeling and Enforcing Advanced Access Control Policies in Healthcare Systems with SECTET.," *In MOTHIS '07: MODELS 2007, Nashville, USA,* 2007.

[139] M. Memon, "Security Modeling with Pattern Refinement for Security-as-a-Service Architecture," *Phd Dissertation, University of Insbruck, Austria, ,* January 2011.

[140] OMG, "Service oriented architecture Modeling Language (SoaML) – Specification for the UML Profile and Metamodel for Services (UPMS)," *http://www.uio.no/studier/emner/matnat/ifi/INF5120/v10/undervisningsmateriale/09-12-09-SoaML.pdf (Date Accesed 21-11-2011),* 2008.

[141] E. Bertino*, et al.*, "Access Control and Authorization Constraints for WS-BPEL," in *International Conference on Web Services, ICWS '06.*, 2006, pp. 275-284.

[142] OMG, "UML 2.0 Specification," *http://www.cs.uga.edu/~kochut/Teaching/x050/UML-2.0-Infrastructure.pdf (Date Accesed 19-01-2012),* 2003.

[143] MagicDraw, "MagicDraw UML Modeling Tool," *https://www.magicdraw.com/,* (Date Accesed 27-07-2011).

[144] M. A. Khurshid, "HL7 V3 Laboratory Messaging using SOA Infrastructure," *Master Thesis, National University of Science and Technology, Pakistan,* April 2011.

[145] P. W. Ron Ten-Hove, "Java™ Business Integration (JBI) 1.0, Final Release," *http://download.oracle.com/otndocs/jcp/jbi-1.0-fr-eval-oth-JSpec/ (Date Accesed 19-08-2011),* August 17, 2005.

[146] Wikipedia, "Application Server," *http://en.wikipedia.org/wiki/Application_server (Date Accesed 30-10-2011).*

[147] T. A. S. Foundation, "Apachi Tomcat," *http://tomcat.apache.org/ (Date Accesed 11-10-2011).*

[148] M. Shaw, "What Makes Good Research in Software Engineering," *International Journal of Software Tools for Technology Transfer, vol. 4, no. 1, pp. 1-7,* 2002.

[149] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering,* vol. 14, pp. 131-164, 2009.

[150] C. B. Seaman, "Qualitative methods in empirical studies of software engineering," *IEEE Transactions on Software Engineering,* vol. 25, pp. 557-572, 1999.

[151] J. S. Steve Easterbrook, Margaret-Anne Storey, Daniela Damian, "Selecting Empirical Methods for Software Engineering Research Guide to Advanced Empirical Software Engineering," F. Shull, *et al.*, Eds., ed: Springer London, 2008, pp. 285-311.

[152] C. T. Abbas Tashakkori, "Mixed Methodology Combining Qualitative and Quantitative Approaches," *SAGE Publication, Inc.,* 1998.

[153] O. A. Leech Nancy, "A typology of mixed methods research designs," *Quality & Quantity,* vol. 43, pp. 265-275, 2009.

[154] C. H. Loraine Blaxter, Malcolm Tight, "How to Research, Second Edition," *Open University Press, Buckingham . Philadelphia.*

[155] C. Dawson, "In Practical Research Methods," *How To Books Ltd, 3 Newtec Place, Magdalen Road, Oxford OX4 1RE. United Kingdom.,* 2002.

[156] J. W. Cresswell, "Research Design: Quantitative, Qualitative and Mix Methods Approaches. 3rd Ed.," *SAGE Publication, Inc.,* 2009.

[157] R. K. Yin, "Case study Research Design and Methods," *3rd Ed. Thousand Oaks, CA:Sage,* 2003.

[158] P. J. M. Verschuren, "Case study as a research strategy: Some ambiguities and opportunities," *International Journal of Social research Methodology,* vol. 6, pp. 121-139.

[159] R. O'Brien, "An Overview of the Methodological Approach of Action Research," *http://www.web.ca/robrien/papers/arfinal.html (Date Accessed 23-10-2012),* 1998.

[160] R. D. E. Gerald I. Susman, "An Assessment of the Scientific Merits of Action Research," *Administrative Science Quarterly,* vol. 23, pp. 582-603, 1978.

[161] A. S. Barney Glaser, "The Discovery of Grounded Theory: Strategies for Qualitative Research," *New York: Aldine de Gruyter,* 1967.

[162] L. C. Burke Johnson, "Educational Research Quantitative, Qualitative and Mixed Approaches Fourth Edition," *SAGE Publications, Inc,* 2011.

[163] M. Q. Patton, "Qualitative evaluation and research methods (2nd ed.)," *Newbury Park, CA: Sage Publications, Inc.,* 1990.

[164] T. N. Basit, "Conducting Research in Educational Contexts," *Continum International Publishing Group,* 2010.

[165] P. V. Maxine Offredy, "Developing a Healthcare Research Proposal: An Interactive Student Guide " *Wiley-Blackwell,* 2010.

[166] J. W. Cresswell, "Research Design: Quantitative, Qualitative and Mix Methods Approaches' (2nd Ed)," *Sage Publications: Thousand Oaks, Ca.,* 2003.

[167] A. Bryman, "Quantity and Quality in Social Research," *Routledge, London,* 1992.

[168] B. P. H. Denise F. Polit, "Nursing Research: Principles and Methods, 6th Edition," *Philadelphia, Lippincott,* 1999.

[169] E. M. María Lázaro, "An Approach to the Integration of Qualitative and Quantitative Research Methods in Software Engineering Research," *2nd Internatinal Workshop on Philosophical Foundations of Information Systems Engineering (PHISE' 06), LNCS. Springer-Verlag, Berlin.,* 2006.

[170] J. D. Murray Wood, James Miller, Marc Roper, "Multi-method research: An empirical investigation of object-oriented technology," *Journal of Systems and Software,* vol. 48, pp. 13-26, 1999.

[171] V. J. C. Jennifer C. Greene, Wendy F. Graham, "Toward a Conceptual Framework for Mixed-Method Evaluation Designs," *Education Evaluation and Policy Analysis,* vol. 11, pp. 255-274, 1989.

[172] J. C. Greene, "Mixed Methods in social Inquiry," *John Wiey and Sons,* 2007.

[173] T. D. Dag I. K. Sjoberg, Magne Jorgensen "The Future of Empirical Methods in Software Engineering Research," *IEEE Conference on Future of Software Engineering (FOSE 07),* 2007.

[174] M. A. Mehtab Alam, Maqbool Hussain, Hafiz Farooq Ahmad, "HL7 Laboratory Messaging Through SOA Infrastructure (HLMSI)," *10th International HL7 Interoperability Conference (IHIC), Kyoto, Japan.,* 8-11 May, 2009.

[175] M. Kasunic, "Designing an Effective Survey," *Carnegie Mellon Software Engineering Institute,* 2005.

[176] B. Kitchenham, "DESMET: A method for evaluating Software Engineering methods and tools," *Technical Report TR96-09, Department of Computer Science University of Keele, Keele, Staffordshire, ST5 5BG, U.K.,* vol. http://www.osel.co.uk/desmet.pdf (Date Accesed: 29-09-2012), 1996.

[177] J. E. McGrath, "Dilemmatic: The studies of the research choice and dilemma' Judgement Call in Research," *Kulka, R. A. Ed. Beverly Hills: Sage Publication pp 69-102,* 1982.

[178] S. Mathison, "Why Triangulate?," *Educational Researcher,* vol. 17, pp. 13-17, 1998.

[179] J. M. Morse, "Approaches to Qualitative-Quantitative Methodologies Triangulation," *Nursing Research,* vol. 40, pp. 120-123, 1991.

[180] M. J. Lars Bratthall, "Can you Trust a Single Data Source Exploratory Software Engineering Case Study?," *Empirical Softw. Engg.,* vol. 7, pp. 9-26, 2002.

[181] V. L. P. C. John W. Cresswell, "Designing and Conducting Mixed Methods Research," *SAGE Publication, Inc.,* 2010.

[182] Z. F. Turner M., Kotsiopoulos I., Russell M., Budgen D., Bennett K., Brereton P., Keane J., Layzell P., Rigby M., "Using Web service technologies to create an information broker: an experience report," in *26th International Conference on Software Engineering, ICSE 2004. Proceedings.* , 2004, pp. 552-561.

[183] P. K. Olaf Zimmermann, Clive Gee, "Elements of Service-Oriented Analysis and Design, An interdisciplinary modeling approach for SOA projects," *http://www.ibm.com/developerworks/library/ws-soad1/ (Date Accesed 14-11-2011),* June 2004.

[184] J. A. Mark Endrei, Ali Arsanjani, Sook Chua, Philippe Comte, Pål Krogdahl, Min Luo, Tony Newling, "Patterns: Service-Oriented Architecture and Web Services," *IBM RedBooks, PP 83-102,* April 2004.

[185] H. I. a. OMG., "Healthcare Services Specication Project (HSSP)," *http://hssp.wikispaces.com/ (Date Accessed May 2011),* October 2004.

[186] F.-M. E. Rodríguez Alfonso, Piattini Mario, "Towards a UML 2.0 Extension for the Modeling of Security Requirements in Business Processes," in *Trust and Privacy in Digital Business*, ed, 2006, pp. 51-61.

[187] U. Sekaran, "Research Methods For Business: A Skill-Building Approach, 4 th Edition.," *Wiley, New York, NY,* 2003.

[188] C. E. Osborn, "Statistical Applications For Health Information Management," *2nd ed. Jones and Bartlett Publisher, Inc.,* 2006.

[189] R. Likert, "A Technique for the Measurement of Attitudes," *Archives of Psychology 140: pp 1–55.,* 1932.

[190] Glenn, "Likert scale & surveys – best practices," *http://intelligentmeasurement.wordpress.com/2007/11/20/likert-scale-surveys-best-practices/ (Date Accessed 25-09-2012),* 2007.

[191] G. Canny, "5 point Vs. 6 point likert scale," *http://geocan.wrytestuff.com/swa69773.htm (Date Accesed 25-09-2012),* 2006.

[192] R. H. Jill Collis, "Business Research: A Practical Guide for Undergraduate and Postgraduate Students," *Palgrave Macmillan, 3rd edition,* 2009.

[193] E. R. Babbie, "Survey Research Methods. ," *Wadsworth Publishing, 2nd edition,* 1990.

[194] A. B. Michael S. Lewis-Beck, Tim Futing Liao, "The SAGE encyclopedia of SOCIAL SCIENCES RESEARCH METHODS . Volume 1.," *SAGE Publication, Inc.,* 2004.

[195] N. K. L. Denzin, Yvonna S., "The Sage Handbook of Qualitative Research (3rd ed.)," *Thousand Oaks, CA: Sage,* 2005.

[196] K.-C. H. Horng-Jinh Chang, "Determination of Sample Size in Using Central LimitTheorem for Weibull Distribution," *Information and Management Sciences,* vol. 17, Number 3, pp. 31-46, 2006.

[197] P. Swanborn, *Case Study Research: What, Why and How?*: SAGE Publications, 2010.

[198] G. Koch, "Basic Allied Health Statistics and Analysis," *3rd Edition. Cengage Learning,* 2008.

[199] J. Pallant, "SPSS Survival Manual: A Step by Step Guide to Data Analysis Using SPSS for Windows (2nd Ed.)," *Maidenhead, Open University Press,* 2005.

[200] U. o. N. E. School of Psycology, "Research Methods and Statistics, PESS202, Lecture and Commentary Notes," *http://www.une.edu.au/WebStat/unit_materials/c4_descriptive_statistics/introduction.html (Date Accesed 30-09-2012),* 2012.

[201] A. C. Arun Kumar, "Descriptive Statistics (Statistical Methods). Fourth Edition," *Krishnan's Text Book in Statistics.,* 2010.

[202] P. Patel, "Introduction to Quantitative Methods," *Empirical Law Seminar,* 2009.

[203] A. W. Röhm*, et al.*, "A Language for Modelling Secure Business Transactions," 1999.

[204] Y. D. Yida Tao, Tao Xie, Dongmei Zhang, Sunghun Kim, "How Do Software Engineers Understand Code Changes ? - An Exploratory Study in Industry," *ACM SIGSOFT-2012, FSE-20, 20th ACM symposium on the foundation of software engineering. Cary, North Carolina, USA,* 2012.

[205] D. I. K. Sjoeberg, Hannay, J. E., Hansen, O., Kampenes, V. B., Karahasanovic, A., Liborg, N. K., Rekdal, A. C., "A survey of controlled experiments in

software engineering," *IEEE Transactions on Software Engineering,* vol. 31, pp. 733-753, 2005.

[206] UML-Diagrame.org, "UML Profile Diagram," *http://www.uml-diagrams.org/profile-diagrams.html (Date Accesed 18-11-2011),* 2010.

[207] H. I. a. M. S. S. (HIMSS), "Privacy and Security," *http://www.himss.org/ASP/topics_privacy.asp (Date Acceded 03-08-2012),* 2009.

[208] P. R. Clearinghouse, "Medical Records Privacy," *http://www.privacyrights.org/fs/fs8-med.htm (Date Accesed 03-08-2012),* 2009.

[209] booz&co, "Information Technology Impact on the Healthcare Industry," *http://www.booz.com/me/home/what_we_think/40007409/40007869/42008005, May202008 (Date Accessed 24-05-2011),* 2008.

[210] C.-I. Community, "Medical Device Integration: References, Resources, and Standards," *http://www.ceitcollaboration.org/docs/GuidanceRecognizedStandards.pdf (Date Accesed 20-11-2011),* 2010.

[211] HL7, " Hl7 v3 guide.," *http://www.hl7.org/v3ballot/html/help/v3guide/v3guide.html (Date Accesed 20-11-2011),* 2010.

[212] B. Company, "Information technology impact on the healthcare industry.," *http://www.booz.com/me/home/what_we_think/40007409/40007869/42008005, May202008* 2008 (Date Accessed 24-05-2011).

[213] D. Peel, "Your Medical Record are't Secure," *The Wallstreet Journal,* March 23, 2010.

[214] S. H. Nancy Orvis, "SOA in Healthcare: Value in a Time of Change," *http://www.omg.org/news/meetings/workshops/SOA-HC/presentations-09/03-18_Hufnagel-Orvis.pdf (Date Accesed 20-11-2011),* 2009.

[215] G. M. Kart F., Moser L. E., Melliar-Smith P. M., "A Distributed e-Healthcare System Based on the Service Oriented Architecture," in *IEEE International Conference on Services Computing, SCC 2007.* , 2007, pp. 652-659.

[216] A. M. Hafner Michael, Breu Ruth, "Towards a MOF/QVT-Based Domain Architecture for Model Driven Security," 2006.

[217] OMG, "Service Functional Model Specification- Entity Identification Service (EIS)," *http://hssp-eis.wikispaces.com/ (Date Accesed 11-08-2011),* vol. Version 0.997, July 17 2006.

LIST OF PUBLICATIONS

Journal Publications

1. Saleem, M. Q., J. Jaafar, Hassan, M. F. "Secure Business Process Modelling of SOA Applications using UML-SOA-Sec". International Journal of Innovative Computing, Information and Control (IJICIC), Vol. 8, No. 4, PP 2729-2746, April 2012.
2. Saleem, M. Q., J. Jaafar, Hassan, M. F. "*Model-based Security Engineering of SOA System Using Security Intent DSL*", International Journal of New Computer Architectures and Their Applications, Vol. 1, No. 2, 2011.
3. Saleem, M. Q., J. Jaafar, Hassan, M. F. "*A Domain Specific Language for Modelling Security Requirements in Business Process Model for SOA Systems*". Advances in Information Sciences and Service Sciences (AISS), an International Journal of Research and Innovation. Vol. 4, No. 1, PP. 353-362, 2012.
4. Saleem, M. Q., J. Jaafar, Hassan, M. F. "A Framework for Model Driven Development of Secure Web Services Composition". Advances in Information Sciences and Service Sciences (AISS), an International Journal of Research and Innovation. Vol. 4 No. 9. PP. 67-78, 2012.
5. Saleem, M. Q., J. Jaafar, Hassan, M. F. "Model-based Security Engineering of SOA Systems Using Modified UML-SOA-Sec". Advances in Information Sciences and Service Sciences (AISS), an International Journal of Research and Innovation. Vol. 4 No. 9. PP. 79-88, 2012.

Conference Publications

1. Saleem, M. Q., J. Jaafar, Hassan, M. F. "Security Modelling Along Business Process Model of SOA Systems Using Modified "UML-SOA-Sec"". IEEE International Conference on Computer and Information Sciences, Kuala Lumpure Malaysia. (ICCIS-2012).
2. Saleem, M. Q., J. Jaafar, Hassan, M. F. "Security Modelling of SOA Systems using Security Intents DSL" Springer LNCS Series, 2nd International

Conferences on Software Engineering and Computer System, (ICSECS 2011), Pahang, Malaysia.

3. Saleem, M. Q., J. Jaafar, Hassan, M. F. "*Model Driven Security Framework for Definition of Security Requirements for SOA systems*" IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE 2010), Kuala Lumpure, Malaysia.
4. Saleem, M. Q., J. Jaafar, Hassan, M. F. "*Model driven security frameworks for addressing security problems of Service Oriented Architecture*" IEEE International Conference, International Symposium in Information Technology (ITSim), 2010, Kuala Lumpure, Malaysia.

APPENDIX A
SURVEY QUESTIONNAIRE

A Comparative Analysis of the different approaches used in different DSLs to annotate the Security in the Business Process Model.

The Objective of the survey is to get your feedback on the different approaches used in different Domain Specific Languages presented by different researchers for security annotation in the Business Process Model.

It will take approximately 5-10 minutes to complete the questionnaire. Your participation in this study is completely voluntary. The survey is completed anonymously and all the data collected in this study will be kept confidential. Your responses will not be passed on to any third party and will only be used for academic research.

If you would like further information about the study please contact me at following:

Muhammad Qaiser Saleem

PhD Student in IT, Department of Computer and Information Sciences,

University Technology PETRONAS (UTP) Bandar Seri Iskandar, 31750, Tronoh Perak, Malaysia

+60 125905242

qaiser_saleem73@hotmail.com

Please provide your feedback by filling the following questionnaire against each researcher.

Researcher Name:- Michal Hafner and Ruth Breu

S/No	Questions	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	The model is messy as it contains many diagrams					
2	The model has abundant Technical details					
3	I can easily add security objectives in the model					
4	The security icons make it easier for me to add security objectives in the model					
5	The numbers of security objectives present in the security DSL are sufficient for SOA environment					

Researcher Name:- Mukhtar Memon

S/No	Questions	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	The model is messy as it contains many diagrams					
2	The model has abundant Technical details					
3	I can easily add security objectives in the model					
4	The security icons make it easier for me to add security objectives in the model					
5	The numbers of security objectives present in the security DSL are sufficient for SOA environment					

Researcher Name:- Alfonso Rodriguez et al.

S/No	Questions	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	The model is messy as it contains many diagrams					
2	The model has abundant Technical details					
3	I can easily add security objectives in the model					
4	The security icons make it easier for me to add security objectives in the model					
5	The numbers of security objectives present in the security DSL are sufficient for SOA environment					

Researcher Name:- Christian Wolter et al.

S/No	Questions	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	The model is messy as it contains many diagrams					
2	The model has abundant Technical details					
3	I can easily add security objectives in the model					
4	The security icons make it easier for me to add security objectives in the model					
5	The numbers of security objectives present in the security DSL are sufficient for SOA environment					

Researcher Name:- Michal Menzel et al.

S/No	Questions	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	The model is messy as it contains many diagrams					
2	The model has abundant Technical details					
3	I can easily add security objectives in the model					
4	The security icons make it easier for me to add security objectives in the model					
5	The numbers of security objectives present in the security DSL are sufficient for SOA environment					

Researcher Name:- Muhammad Qaiser Saleem (Proposed)

S/No	Questions	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	The model is messy as it contains many diagrams					
2	The model has abundant Technical details					
3	I can easily add security objectives in the model					
4	The security icons make it easier for me to add security objectives in the model					
5	The numbers of security objectives present in the security DSL are sufficient for SOA environment					

Thank for your feed back

APPENDIX B

LIST OF RESPONDENTS OF THE SURVEY

S/No	Affiliation	Qualification	Designation	Mode of Data Collection	Knowledge of UML or BPMN	Familiar with Security Objectives
1	Computer and Information Sciences Department, Universiti Tecknologi PETRONAS, Malaysia.	Master	PhD Scholar	Personally Administered Questionnaire	Yes	Yes
2	As Above	Master	PhD Scholar	As Above	Yes	Yes
3	As Above	Master	PhD Scholar	As Above	Yes	Yes
4	As Above	Master	PhD Scholar	As Above	Yes	Yes
5	As Above	Master	PhD Scholar	As Above	Yes	Yes
6	As Above	Master	PhD Scholar	As Above	Yes	Yes
7	As Above	Master	PhD Scholar	As Above	Yes	Yes
8	As Above	Master	PhD Scholar	As Above	Yes	Yes
9	As Above	Master	PhD Scholar	As Above	Yes	Yes
10	As Above	Master	PhD Scholar	As Above	Yes	Yes
11	As Above	Master	PhD Scholar	As Above	Yes	Yes

12	As Above	Master	PhD Scholar	As Above	Yes	Yes
13	As Above	Master	PhD Scholar	As Above	Yes	Yes
14	As Above	Master	PhD Scholar	As Above	Yes	Yes
15	As Above	Master	PhD Scholar	As Above	Yes	Yes
16	As Above	Master	PhD Scholar	As Above	Yes	Yes
17	As Above	Master	PhD Scholar	As Above	Yes	Yes
18	As Above	Master	PhD Scholar	As Above	Yes	Yes
19	As Above	Master	PhD Scholar	As Above	Yes	Yes
20	As Above	Master	PhD Scholar	As Above	Yes	Yes
21	As Above	Master	PhD Scholar	As Above	Yes	Yes
22	As Above	Master	PhD Scholar	As Above	Yes	Yes
23	True Meridian Private Limited, Islamabad, Pakistan. http://www.truemeridian.com/Company.html	Master	Software Engineer	Through Email	Yes	Yes
24	As Above	Bachelor	Software Engineer	Through Email	Yes	Yes
25	As Above	Master	Project Manager	Through Email	Yes	Yes
26	As Above	Master	Project Manager	Through Email	Yes	Yes
27	As Above	Master	Senior Software Engineer	Through Email	Yes	Yes

28	As Above	Bachelor	Software Engineer	Through Email	Yes	Yes
29	As Above	Master	Software Engineer	Through Email	Yes	Yes
30	As Above	Master	Senior Software Engineer	Through Email	Yes	Yes

APPENDIX C
RESPONDENT'S FEEDBACK

Feedback of the respondents is organized in tables according to model of a particular researcher.

Michal Hafner and Ruth Breu:

S/No	Questions	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	The model is messy as it contains many diagrams	8	10	7	4	1
2	The model has abundant Technical details	7	12	5	5	1
3	I can easily add security objectives in the model	3	10	5	9	3
4	The security icons make it easier for me to add security objectives in the model	No icons are provided				
5	The numbers of security objectives present in the security DSL are sufficient for SOA environment	3	11	2	9	5

Mukhtiar Memon:

S/No	Questions	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	The model is messy as it contains many diagrams	7	11	5	6	1
2	The model has abundant Technical details	6	12	5	6	1
3	I can easily add security objectives in the model	4	12	4	8	2
4	The security icons make it easier for me to add security objectives in the model	No icons are provided				
5	The numbers of security objectives present in the security DSL are sufficient for SOA environment	2	6	2	14	6

Alfonso Rodriguez et al.:

S/No	Questions	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	The model is messy as it contains many diagrams	1	5	4	14	6
2	The model has abundant Technical details	2	5	3	14	6
3	I can easily add security objectives in the model	6	14	4	5	1
4	The security icons make it easier for me to add security objectives in the model	8	14	3	3	2
5	The numbers of security objectives present in the security DSL are sufficient for SOA environment	4	13	2	9	2

Christian Wolter et al.

S/No	Questions	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	The model is messy as it contains many diagrams	5	14	5	5	1
2	The model has abundant Technical details	6	14	3	4	3
3	I can easily add security objectives in the model	4	11	2	7	6
4	The security icons make it easier for me to add security objectives in the model	6	15	1	7	1
5	The numbers of security objectives present in the security DSL are sufficient for SOA environment	5	14	4	7	0

Michal Menzel et al.

S/No	Questions	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	The model is messy as it contains many diagrams	1	5	5	16	3
2	The model has abundant Technical details	5	16	2	6	1
3	I can easily add security objectives in the model	5	15	4	5	1
4	The security icons make it easier for me to add security objectives in the model	6	15	4	4	1
5	The numbers of security objectives present in the security DSL are sufficient for SOA environment	6	14	3	5	2

Muhammad Qaiser Saleem (Proposed):

S/No	Questions	Strongly Agree	Agree	Neutr al	Disagree	Strongly Disagree
1	The model is messy as it contains many diagrams	1	3	4	15	7
2	The model has abundant Technical details	1	2	2	17	8
3	I can easily add security objectives in the model	7	15	4	3	1
4	The security icons make it easier for me to add security objectives in the model	8	15	2	3	2
5	The numbers of security objectives present in the security DSL are sufficient for SOA environment	8	16	4	2	0

APPENDIX D
SAMPLE CODE

Code of the implementation is represented in this chapter regarding the composite service in section 1 and WSDL description of the basic services in section 2.

Section 1 : Code Description Of Composite Services

BusinessService and SecurityService are the two basic services as described in the UML deployment diagram in Figure 4.5. Their description is provided in this section.

1.1 Code For The BusinessServices

```
package com.hl7.service.web;

import com.hl7.service.BusinessServiceImpl;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;

/**
 *
 *
 */
@WebService
public class BusinessService {

    private final BusinessServiceImpl __businessServiceImpl = new BusinessServiceImpl();

    @WebMethod(operationName = "connect")
    public boolean connect(@WebParam(name = "sid") String sid) {
        return __businessServiceImpl.connect(sid);
    }

    @WebMethod(operationName = "releaseResources")
    public boolean releaseResources(@WebParam(name = "sid") String sid) {
        return __businessServiceImpl.releaseResources(sid);
    }

    @WebMethod(operationName = "searchPatientById")
    public boolean searchPatientById(@WebParam(name = "sid") String sid, @WebParam(name = "pid") int pid) {
        return __businessServiceImpl.searchPatientById(sid, pid);
    }

    @WebMethod(operationName = "eis_Operation")
    public boolean eis_Operation(@WebParam(name = "sid") String sid) {
        return __businessServiceImpl.eis_Operation(sid);
    }
}

```

1.2 Code For The SecurityServices

```
package com.hl7.service.web;

import com.hl7.service.SecurityServiceImpl;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;

/**
 *
 *
 */
```

```
@WebService
public class SecurityService {

    private static final com.hl7.service.SecurityService __securityServiceImpl = SecurityServiceImpl.getInstance();

    @WebMethod(operationName = "authenticate")
    public String authenticate(@WebParam(name = "userId") String userId, @WebParam(name = "password") String password) {
        return __securityServiceImpl.authenticate(userId, password);
    }

    @WebMethod(operationName = "getPermission")
    public boolean getPermission(@WebParam(name = "sid") String sid, @WebParam(name = "methodName") String methodName, @WebParam(name = "serviceName") String serviceName) {
        return __securityServiceImpl.getPermission(sid, methodName, serviceName);
    }
}

```

Section 2: WSDL Description of EIS Composite Application Business Services

BusinessService is composed of five basic EIS services naming Database service, EIS_EntryPoint service, Message service, Parser service and XEIS service. WSDL descriptions of these five basic EIS services are provided in below.

2.1 WSDL Description of EIS Entry Point Service

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="EIS_Entry_Point" targetNamespace="http://j2ee.netbeans.org/wsdl/EIS_Entry_Point"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://j2ee.netbeans.org/wsdl/EIS_Entry_Point"
```

```
xmlns:plnk="http://docs.oasis-open.org/wsbpel/2.0/plnktype"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
  <types/>
  <message name="EIS_Entry_PointOperationRequest">
    <part name="part1" type="xsd:string"/>
  </message>
  <message name="EIS_Entry_PointOperationResponse">
    <part name="part1" type="xsd:string"/>
  </message>
  <portType name="EIS_Entry_PointPortType">
    <operation name="EIS_Entry_PointOperation">
      <input name="input1" message="tns:EIS_Entry_PointOperationRequest"/>
      <output name="output1"
message="tns:EIS_Entry_PointOperationResponse"/>
    </operation>
  </portType>
  <binding name="EIS_Entry_PointBinding"
type="tns:EIS_Entry_PointPortType">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="EIS_Entry_PointOperation">
      <soap:operation/>
      <input name="input1">
        <soap:body use="literal"
namespace="http://j2ee.netbeans.org/wsdl/EIS_Entry_Point"/>
      </input>
      <output name="output1">
        <soap:body use="literal"
namespace="http://j2ee.netbeans.org/wsdl/EIS_Entry_Point"/>
      </output>
    </operation>
  </binding>
  <service name="EIS_Entry_PointService">
    <port name="EIS_Entry_PointPort" binding="tns:EIS_Entry_PointBinding">
```

```
            <soap:address location="http://localhost:${HttpDefaultPort}/EIS_Entry_PointService/EIS_Entry_PointPort"/>
        </port>
    </service>
    <plnk:partnerLinkType name="EIS_Entry_Point">
        <!-- A partner link type is automatically generated when a new port type is added. Partner link types are used by BPEL processes.
In a BPEL process, a partner link represents the interaction between the BPEL process and a partner service. Each partner link is associated with a partner link type.
A partner link type characterizes the conversational relationship between two services. The partner link type can have one or two roles.-->
        <plnk:role name="EIS_Entry_PointPortTypeRole" portType="tns:EIS_Entry_PointPortType"/>
    </plnk:partnerLinkType>
</definitions>
```

2.2 WSDL Description of Database Service

```
<?xml version="1.0" encoding="UTF-8"?>
 <definitions name="DATABASE" targetNamespace="http://j2ee.netbeans.org/wsdl/DATABASE"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://j2ee.netbeans.org/wsdl/DATABASE" xmlns:plnk="http://docs.oasis-open.org/wsbpel/2.0/plnktype" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
    <types/>
    <message name="DATABASEOperationRequest">
        <part name="part1" type="xsd:string"/>
    </message>
     <message name="DATABASEOperationResponse">
        <part name="part1" type="xsd:string"/>
```

```
</message>
<portType name="DATABASEPortType">
    <operation name="DATABASEOperation">
        <input name="input1" message="tns:DATABASEOperationRequest"/>
        <output name="output1" message="tns:DATABASEOperationResponse"/>
    </operation>
</portType>
<binding name="DATABASEBinding" type="tns:DATABASEPortType">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="DATABASEOperation">
        <soap:operation/>
        <input name="input1">
            <soap:body use="literal" namespace="http://j2ee.netbeans.org/wsdl/DATABASE"/>
        </input>
        <output name="output1">
            <soap:body use="literal" namespace="http://j2ee.netbeans.org/wsdl/DATABASE"/>
        </output>
    </operation>
</binding>
<service name="DATABASEService">
    <port name="DATABASEPort" binding="tns:DATABASEBinding">
        <soap:address location="http://localhost:${HttpDefaultPort}/DATABASEService/DATABASEPort"/>
    </port>
</service>
<plnk:partnerLinkType name="DATABASE">
    <!-- A partner link type is automatically generated when a new port type is added. Partner link types are used by BPEL processes.
In a BPEL process, a partner link represents the interaction between the BPEL process and a partner service. Each partner link is associated with a partner link type.
```

```
A partner link type characterizes the conversational relationship between two services. The partner link type can have one or two roles.-->
        <plnk:role name="DATABASEPortTypeRole" portType="tns:DATABASEPortType"/>
    </plnk:partnerLinkType>
</definitions>

```

2.3 WSDL Description of Message Service

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="MESSAGE" targetNamespace="http://j2ee.netbeans.org/wsdl/MESSAGE"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://j2ee.netbeans.org/wsdl/MESSAGE" xmlns:plnk="http://docs.oasis-open.org/wsbpel/2.0/plnktype" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
    <types/>
    <message name="MESSAGEOperationRequest">
        <part name="part1" type="xsd:string"/>
    </message>
    <message name="MESSAGEOperationResponse">
        <part name="part1" type="xsd:string"/>
    </message>
    <portType name="MESSAGEPortType">
        <operation name="MESSAGEOperation">
            <input name="input1" message="tns:MESSAGEOperationRequest"/>
            <output name="output1" message="tns:MESSAGEOperationResponse"/>
        </operation>
    </portType>
    <binding name="MESSAGEBinding" type="tns:MESSAGEPortType">
        <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="MESSAGEOperation">
            <soap:operation/>
            <input name="input1">
                <soap:body use="literal" namespace="http://j2ee.netbeans.org/wsdl/MESSAGE"/>
            </input>
            <output name="output1">
                <soap:body use="literal" namespace="http://j2ee.netbeans.org/wsdl/MESSAGE"/>
            </output>
        </operation>
    </binding>
    <service name="MESSAGEService">
        <port name="MESSAGEPort" binding="tns:MESSAGEBinding">
            <soap:address location="http://localhost:${HttpDefaultPort}/MESSAGEService/MESSAGEPort"/>
        </port>
    </service>
    <plnk:partnerLinkType name="MESSAGE">
        <!-- A partner link type is automatically generated when a new port type is added. Partner link types are used by BPEL processes.
In a BPEL process, a partner link represents the interaction between the BPEL process and a partner service. Each partner link is associated with a partner link type.
A partner link type characterizes the conversational relationship between two services. The partner link type can have one or two roles.-->
        <plnk:role name="MESSAGEPortTypeRole" portType="tns:MESSAGEPortType"/>
    </plnk:partnerLinkType>
</definitions>

```

2.4 WSDL Description of Parser Service

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="PARSER" targetNamespace="http://j2ee.netbeans.org/wsdl/PARSER"
```

```
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://j2ee.netbeans.org/wsdl/PARSER" xmlns:plnk="http://docs.oasis-open.org/wsbpel/2.0/plnktype" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
<types/>
<message name="PARSEROperationRequest">
    <part name="part1" type="xsd:string"/>
</message>
<message name="PARSEROperationResponse">
    <part name="part1" type="xsd:string"/>
</message>
<portType name="PARSERPortType">
    <operation name="PARSEROperation">
        <input name="input1" message="tns:PARSEROperationRequest"/>
        <output name="output1" message="tns:PARSEROperationResponse"/>
    </operation>
</portType>
<binding name="PARSERBinding" type="tns:PARSERPortType">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="PARSEROperation">
        <soap:operation/>
        <input name="input1">
            <soap:body use="literal" namespace="http://j2ee.netbeans.org/wsdl/PARSER"/>
        </input>
        <output name="output1">
            <soap:body use="literal" namespace="http://j2ee.netbeans.org/wsdl/PARSER"/>
        </output>
    </operation>
</binding>
<service name="PARSERService">
    <port name="PARSERPort" binding="tns:PARSERBinding">
```

```
            <soap:address location="http://localhost:${HttpDefaultPort}/PARSERService/PARSERPort"/>
        </port>
    </service>
    <plnk:partnerLinkType name="PARSER">
        <!-- A partner link type is automatically generated when a new port type is added. Partner link types are used by BPEL processes.
In a BPEL process, a partner link represents the interaction between the BPEL process and a partner service. Each partner link is associated with a partner link type.
A partner link type characterizes the conversational relationship between two services. The partner link type can have one or two roles.-->
        <plnk:role name="PARSERPortTypeRole" portType="tns:PARSERPortType"/>
    </plnk:partnerLinkType>
</definitions>

```

2.5 WSDL Description of XEIS Service

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="XEIS" targetNamespace="http://j2ee.netbeans.org/wsdl/XEIS"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://j2ee.netbeans.org/wsdl/XEIS" xmlns:plnk="http://docs.oasis-open.org/wsbpel/2.0/plnktype" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
    <types/>
    <message name="XEISOperationRequest">
        <part name="part1" type="xsd:string"/>
    </message>
    <message name="XEISOperationResponse">
        <part name="part1" type="xsd:string"/>
    </message>
    <portType name="XEISPortType">
        <operation name="XEISOperation">
```

```
            <input name="input1" message="tns:XEISOperationRequest"/>
            <output name="output1" message="tns:XEISOperationResponse"/>
        </operation>
    </portType>
    <binding name="XEISBinding" type="tns:XEISPortType">
        <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="XEISOperation">
            <soap:operation/>
            <input name="input1">
                <soap:body use="literal" namespace="http://j2ee.netbeans.org/wsdl/XEIS"/>
            </input>
            <output name="output1">
                <soap:body use="literal" namespace="http://j2ee.netbeans.org/wsdl/XEIS"/>
            </output>
        </operation>
    </binding>
    <service name="XEISService">
        <port name="XEISPort" binding="tns:XEISBinding">
            <soap:address location="http://localhost:${HttpDefaultPort}/XEISService/XEISPort"/>
        </port>
    </service>
    <plnk:partnerLinkType name="XEIS">
        <!-- A partner link type is automatically generated when a new port type is added. Partner link types are used by BPEL processes.
In a BPEL process, a partner link represents the interaction between the BPEL process and a partner service. Each partner link is associated with a partner link type.
A partner link type characterizes the conversational relationship between two services. The partner link type can have one or two roles.-->
        <plnk:role name="XEISPortTypeRole" portType="tns:XEISPortType"/>
    </plnk:partnerLinkType>
</definitions>
```