

Web Information System Development and Integration by Utilizing Enterprise Architecture and Model Driven Architecture in Educational Institutions

マルディアナ

<https://doi.org/10.15017/1398390>

出版情報：九州大学，2013，博士（工学），課程博士
バージョン：
権利関係：全文ファイル公表済



**Web Information System Development and Integration
by Utilizing Enterprise Architecture and Model Driven
Architecture in Educational Institutions**

Mardiana

**Department of Advanced Information Technology
Graduate School of Information Science and Electrical Engineering
Kyushu University
2013**

Web Information System Development and Integration
by Utilizing Enterprise Architecture and Model Driven
Architecture in Educational Institutions

Name : Mardiana

Student ID : 3IE10039M

Date : July, 2013

Supervisor : Prof. Keijiro Araki

Preface

The main problem currently confronting medium- and large-scale enterprises such as educational institutions concerns the lack of flexibility of their Information Technology (IT) systems. Business processes can change in a matter of days, yet IT architects are unable to keep pace with such rapid changes. An alteration in the business process will influence on the need to develop a new Information System (IS) or change an existing one. Developing or changing an IS must be done within a time period acceptable to the business condition of the institutions. On the other hand, the development of software to support changes in IS is incapable of meeting this need. Another issue that educational institutions in Indonesia must also face concerns inconsistency in the development of IS and lack of integration between an IS with the other. This issue needs to be dealt with immediately to prevent a whole set of more complex problems from emerging. These issues are the underlying rationale for bringing this thesis forward.

This thesis presents a study of the model-driven approach to integrate enterprise and web information systems development in educational institutions. The study seeks to meet the needs of higher educational institutions in Indonesia for the appropriate standards, models and methods for developing and integrating Web Information System (WIS). This thesis provides three key contributions – an IT architecture model, a WIS development method, and an evaluation of WIS model – each of which will be described in chapters 3, 4 and 6 in this thesis. A case study that focuses on the University of Lampung (Unila) will apply the proposed models and method.

In this thesis, the advantages of combining the Enterprise Architecture (EA) and Model Driven Architecture (MDA) concepts will be applied to produce a comprehensive IT architecture

model for a Higher Educational Institution (HEI). This model is necessary to guide activities required for the development of an EA for a HEI, particularly the WIS architecture. For the development and integration of existing WISs, an MDA SOA-based development method is proposed to facilitate the process. This study will then be complemented with the construction of a WIS evaluation model to assess the quality and effectiveness of WIS. I believe that the integration of these three highly important concepts – EA, MDA and SOA-, gathered in one consistent WIS development process, will provide practitioners with many benefits.

The initial motivation for conducting this research comes from my experience gained from being involved in work related to IT and the development of IS at Unila's Computer Center. Various issues and solutions are interesting to observe and examine, and indirectly contribute to the preparing of this thesis. I have researched and explored further into the research work at the Lab. Advanced Software Engineering, Dep. of Advanced Information Technology, Kyushu University until the completion of this thesis.

This study mainly presents a general (national) standard as the reference approach for analyzing, integrating and developing WIS at HEI. It is expected to contribute for the development and integration of existing WISs in higher education institutions in Indonesia. The WIS planned by HEI can therefore meet its needs, and can also be developed and evaluated in an easy, quick and efficient way.

Acknowledgements

All the praises and thank be to Allah Subhanahu wa-ta'ala, above all the gracious that was given to me. Even though the gratitude is not expressible enough, I would like to express these acknowledgement as my appreciations to some persons and institutions which have supported during the study in Kyushu University.

With my best regards, I would like to express my profound gratitude with full respect to my supervisor Prof. Keijiro Araki for giving me opportunity to study in Kyushu University, for always helping me with his invaluable advices and suggestions along way. I appreciate his insightful guidance and encouragement that inspire me at every step of my study. Without him this thesis would have not yet been achieved. Also, during studying and doing research, I was given many opportunities to gain experiences by attending to international conferences, and a workshop. I believe that these valuable experiences will help my work as a lecturer and a researcher in the future.

With my great honor to have Assoc. Prof. Shigeru Kusakabe, Assoc. Prof. Tsunenori Mine, and Assoc. Prof. Daisuke Ikeda serving as advisory committee members, I would like to thank for their critical questions, constructive comments and valuable suggestions on my PhD thesis. I received many suggestions and directions about my research during advisory committee meetings.

Sincere thanks to Dr. Yoichi Omori and Ms. Miki Nakagami for always helping me from the beginning of my study in Araki-Kusakabe laboratory. Thanks to Shinya Yamada, Mitsuhide Honda, Kanamaru Shohei, Samih Mohammed and all the members of Araki-Kusakabe laboratory. I appreciate all of their friendship and encouragement. All of our laboratory meeting, cooperative activities, chatting and experiences will always remain in my memory.

Acknowledgments

I owe appreciation to the staff of Kyushu University International Student and Researcher Support Centre, in particular to Hiroko Yamaguchi and Terumi Perry for their helpful support for me and my family regarding the administration works.

I would like to thank to the Department of Electrical engineering University of Lampung, to my colleagues Dr. Ardian Ulvan, Wahyu Eko Sulistiono, M.Sc, M. Komaruddin, M.T (Head of Unila Computer Centre), Denny Chandra S.T, for their helps and discussions regarding practical issues. Thanks also to Dewi Agustina, Dr. Dikpride Despa, Dr. Endah Komalasari and Dr. Yul Martin, for always concern to me during our study in Japan. Thank you for always being there for me, in times of when the research was going to plan, but also in stressful periods.

Many thanks to Dr. Mohammad Fajar, Rimba Whidiana and Nur Aida for many useful discussions and comments. I should also appreciate all staff and friends in Ryugakusei Imajuku Kaikan, all the members of Indonesian Student Association in Fukuoka (PPIF) for helping me a lot in setting up and enlightening my life in Fukuoka.

I am grateful to the Directorate General of Higher Education (DGHE) of the Ministry of National Education of Indonesia that has provided the funding of this study.

Last but not least, I would like to dedicate this work to my husband Mr. S. Rendra Utama, my lovely son Shahzadan Mulla Akhtar, and all my family members (my mothers, sisters and brothers). Thank you for your endless support, love and prayers. I eternally love you all.

Abstract

Web Information System (WIS) for educational institutions is relatively different from other enterprises due to dissimilarities in terms of purpose, problem characteristics and users. Higher Education Institution (HEI) requires the appropriate standards and models to conform in order to establish a WIS architecture that is both suitable and straightforward with regard to its development process and evaluation. This also implies that HEI requires an appropriate approach to integrate all work related to enterprise architecture and the development of its web information systems.

This thesis provides three key contributions: an Information Technology (IT) architecture model, a WIS development method, and a WIS evaluation model that are summarized in the web information system educational framework.

Enterprise Architecture (EA) and Model Driven Architecture (MDA) are concepts for formulating an IT architecture model into activities related to WIS development to meet university needs. Based on an analysis of information technology and information systems of HEIs in Indonesia, an IT architecture model has been proposed according to the situation and needs of national universities. This architecture model is then applied to University of Lampung (Unila, Indonesia) as a case study. The outcome is an IT architecture that can be implemented to support integrated WIS development in Unila. The IT architecture consists of business architecture, information system architecture and technology architecture. The result of this study indicates that the proposed IT architecture model can be a reference for HEIs in order to produce architectures in accordance with the business process and WIS needs.

Several principles and platforms planned within the IT architecture can lead to changes in the existing information systems and technologies. Alterations of existing information systems or

the development of a new information system for a HEI must be done within a time span acceptable to the current business condition. On the other hand, software developments to support information system modifications are still incapable of keeping pace with current needs. In this study, we propose WIS development method based on MDA and Service Oriented Architecture (SOA) approaches. This method defines the WIS development process to automatically obtain web application interface from information system model. Results from this study shed light on how our method can be applied and effectively integrate existing WISs in HEI. We illustrate a development of web application interface by applying the proposed method to integrate two existing WISs at Unila. The implementation of a web application interface (called SIMPEL) has successfully integrated Course Management System into Unila's Academic Information System and work properly. The WIS development process involved in this method allows developers to progress smoothly from the models to the generation of the web applications. Usefulness of the templates to assistance automatically generate PHP files in the form of Model View Controller (MVC), Web Service Description Language (WSDL) and Data Definition Language (DDL) is powerful enough to cover the constraints of time effort and write repetitive code. It can be concluded that the adoption of this method can help developers efficiently develop and increase productivity of WIS.

This thesis is complemented with constructing a new model for evaluating WIS that allows university to assess quality and effectiveness of developed and implemented WIS. The evaluation model is constructed by modifying the concepts of both Technology Acceptance Model (TAM) and IS-Impact due to their inadequate comprehensive evaluation for WIS. The model then applied to assess the quality and effectiveness of SIMPEL in Unila. It described in two parameters: performance evaluation and functionality evaluation. The overall evaluation results show that the evaluation model can be valuable for decision makers in HEI on evaluating the development and implementation of WIS.

Three key aspects - IT architecture model, WIS development method, and WIS evaluation model – incorporated into the WIS education framework are highlighted in this thesis in contributing towards resolving issues related to the development and integration of existing WISs in educational institutions in a way that the planned WIS can effectively meet institutional needs, and can be developed and evaluated in an easy, quick and efficient way.

Abstract

The result of this work is expected to specifically contribute to the advancement of educational institutions in Indonesia, particularly as a reference for developing and integrating WIS.

Key words : Enterprise Architecture, Model Driven Architecture, Service Oriented Architecture, Web Information System Development, Integration, Higher Education Institution

Contents

Preface.....	iii
Acknowledgements.....	v
Abstract	vii
List of Figures	xvii
List of Tables	xxi
Chapter 1 Introduction	1
1.1 Background.....	1
1.2 Problem Statement.....	4
1.3 Research Objectives	5
1.4 Thesis Contribution	6
1.5 Organization of Thesis.....	7
Chapter 2 Key Concepts and Technologies.....	11
2.1 Web Information Systems (WIS).....	11
2.1.1 Methodologies for Web Information Systems.....	12

Contents

2.1.2	Integration and Interoperability Issues of Web Information Systems	14
2.1.3	WIS : The Case of Higher Education Institutions Indonesia	16
2.2	Model Driven Architecture (MDA) and Service Oriented Architecture (SOA)	17
2.2.1	Model Driven Architecture (MDA)	18
2.2.2	SOA Technologies and Web Service	22
2.3	Conceptualization of Information Technology Architecture Model	24
2.3.1	Enterprise Architecture (EA)	24
2.3.2	Enterprise Architecture and related MDA Concepts	27
2.4	Related Work	28
Chapter 3 EA-MDA Model to Resolve IS Characteristic Problems in Educational Institutions		33
3.1	Chapter Introduction	33
3.2	The Current Situation of IS in Educational Institutions in Indonesia	35
3.3	Issues and Challenges	37
3.3.1	Integration	37
3.3.2	Interoperability	38
3.4	WIS Development Strategy	38
3.5	Requirements Analysis of WIS	39
3.5.1	Internal Requirements	39
3.5.2	External Requirements	39
3.6	Architecture Used: Creating the foundation for WIS Development	41

Contents

3.6.1	Enterprise Architecture	42
3.6.2	Support Activities	44
3.6.3	Technology Infrastructure	47
3.6.4	MDA based WIS Development.....	47
3.6.5	IS Implementation	48
3.6.6	IS Evaluation.....	49
3.7	IT Architecture : A Case Study.....	49
3.7.1	Business Architecture	49
3.7.2	Information System Architecture.....	50
3.7.3	Technology Architecture.....	55
3.7.4	Solution Strategy for Bridging the Gap.....	56
3.7.5	Migration and Planning	58
3.7.6	Implementation Strategy.....	58
3.8	Chapter Summary	59
Chapter 4	MDA and Integration of Web Information Systems.....	61
4.1	Chapter Introduction.....	61
4.2	Web Information System Development.....	63
4.2.1	Web Design	63
4.2.2	Data and Web Services as Integration Path.....	64
4.3	WIS Development based on MDA-SOA	66

Contents

4.3.1	The Proposed Method for WIS Development.....	66
4.3.2	WIS Development Process	68
4.3.3	Guideline for WIS development process	82
4.4	Chapter Summary	85
Chapter 5	SIMPEL : A Case Study	87
5.1	Chapter Introduction.....	87
5.2	Existing Web Information Systems	88
5.2.1	Course Management System	89
5.2.2	Academic Information System	90
5.3	The SIMPEL Development	91
5.3.1	Description	91
5.3.2	Functionality	98
5.3.3	Services Categories.....	100
5.3.4	The generated Web Application	104
5.4	The SIMPEL Implementation	111
5.4.1	Demonstration Scenario	111
5.4.2	Evaluation of SIMPEL	115
5.5	Chapter Summary	117
Chapter 6	Evaluation Model for WIS Quality and Effectiveness.....	119
6.1	Chapter Introduction.....	119

Contents

6.2	WIS Evaluation Model	120
6.2.1	A Reference WIS Evaluation Model.....	121
6.2.2	Measurement Dimensions.....	123
6.3	Measurement Results	124
6.3.1	Performance Evaluation	124
6.3.2	Functionality Evaluation	128
6.4	Chapter Summary	129
Chapter 7	Conclusion and Future Work	131
7.1	Conclusion	131
7.2	Future Work.....	134
References		137
Publications.....		145
Index		147
List of Abbreviations.....		149

List of Figures

Figure 1.1 Research Framework.....	6
Figure 2.1 Work environment of web service.....	23
Figure 2.2 Enterprise Architectural Relationships.....	25
Figure 2.3 Modified version of EA concepts and relationships.....	26
Figure 2.4 TOGAF ADM and related MDA concepts.....	28
Figure 3.1 Information Technology Architecture Model for HEI.....	41
Figure 3.2 Unila's business process.....	50
Figure 3.3 Data Architecture.....	51
Figure 3.4 Application Architecture.....	54
Figure 3.5 Proposed technology architecture.....	56
Figure 3.6 Integration Model Architecture.....	58
Figure 4.1 Illustration of web application interface	64
Figure 4.2 Web application interface design	65
Figure 4.3 The WIS development method.....	67
Figure 4.4 Detail of WIS development process.....	69
Figure 4.5 A simplified form of the metamodel.....	70
Figure 4.6 Templates for generating files.....	71
Figure 4.7 An excerpt of one template for the MVC_Model : CRUDModelTemplate.....	73

List of Figures

Figure 4.8 An excerpt of MainCRUDGenerator.java.....	75
Figure 4.9 An excerpt of UserModel.php	75
Figure 4.10 An excerpt of DDLTemplate.....	77
Figure 4.11 An excerpt of WSDLTemplate.....	78
Figure 4.12 An excerpt of UserDatum.wsdl.....	78
Figure 4.13 Structure folders and files generated by the generator code	79
Figure 4.14 MVC pattern of web application.....	81
Figure 4.15 Architecture of the generated web application.....	82
Figure 5.1 Existing WIS in Academic and Knowledge Management	89
Figure 5.2 A Screenshot of MOODLE Unila	90
Figure 5.3 A screenshot of SIAKAD (Academic Information System) Unila	91
Figure 5.4 Overview SIMPEL	92
Figure 5.5 Data flow and user interaction with SIMPEL	94
Figure 5.6 Use Case diagram.....	99
Figure 5.7 Course Enrolment (SIMPEL activities and services)	102
Figure 5.8 Course Registration (SIMPEL activities and services).....	103
Figure 5.9 Student Grouping (SIMPEL activities and services).....	103
Figure 5.10 Learning Strategist (SIMPEL activities and services)	104
Figure 5.11 An example of MVC web application model	105
Figure 5.12 Illustration of Model and Controller Template to generate PHP files	106
Figure 5.13 The CRUD UI for updating the database	107
Figure 5.14 Model for the Registration and Enrolment Course.....	107
Figure 5.15 The CRUD UI for Registration.....	108
Figure 5.16 The WS UI for Course Enrolment	109

List of Figures

Figure 5.17 An excerpt source code for the enrolling course (CourseDatumModel.php)	109
Figure 5.18 Illustration of DDL model, DDLTemplate and SQL File	110
Figure 5.19 Generated files for the SIMPEL application	111
Figure 5.20 Demonstration of the interaction student	113
Figure 5.21 Demonstration of the interaction teacher	114
Figure 5.22 Number of enroll course and active student in course	115
Figure 5.23 Distribution of students in each group.....	116

List of Tables

Table 3.1 Data and reports needed for National Higher Education Database (PDPT).....	40
Table 3.2 Data and reports needed for EPSBED report	40
Table 3.3 Analysis results of planned data (to-be)	50
Table 3.4 Analysis results of planned application (to-be)	52
Table 4.1. MVC Templates.....	72
Table 4.2 DDL Templates.....	72
Table 4.3 WSDL Templates.....	72
Table 4.4 Activity and Output of WIS development process	84
Table 5.1 Integration parts and reasons for integration.....	92
Table 5.2 User Registration	95
Table 5.3 Course Enrolment	96
Table 5.4 Course Registration.....	96
Table 5.5 Student Grouping	97
Table 5.6 Learning Strategist	97
Table 5.7 Grade Calculation	98
Table 5.8 List of use case and related actor	99
Table 5.9 Available Web Services	101
Table 5.10 Created Web services	101

List of Tables

Table 6.1 Dimensions and elements for measuring the WIS	123
Table 6.2 User and related element	124
Table 6.3 Measurement result of SIMPEL	125
Table 6.4 Time development of SIMPEL.....	126
Table 6.5 SIMPEL quality measures	127
Table 6.6 Functionality Evaluation result	128

Chapter 1 Introduction

This chapter introduces the background of this thesis. It presents about a number of problems and shortcomings regarding the architecture, development, and evaluation of WIS. It also identifies integration needs of existing web information systems in educational institutions. The problem statement, research objectives, thesis contribution, as well as the organization of thesis is presented in this chapter.

1.1 Background

Educational institutions play a critical role in society, primarily in developing countries. Educational institutions which in this case refer to Higher Education Institutions have business domains unlike that of business entities or other organizations. Higher Education Institution places more emphasis on the role of Information Technology (IT) in supporting its business processes. Given its scale and complexity, IT management in HEI can be categorized as enterprise-scale data management. As such, particular planning and design is necessary to ensure that IT is applied in accordance with the institution's strategic objective and plan, and that it can be optimally utilized by users.

The purpose of HEIs in Indonesia adheres to the *Tridharma Perguruan Tinggi*¹ (Triple Principles of Higher Education Institutions). These principles are the main functions to be realized by the

¹ Tridharma Perguruan Tinggi is Indonesian terminology meaning Triple Principles of Higher Education Institutions.

every HEI that comprise education, research and community service. Rules of procedure of these principles refer to Law of the Republic of Indonesia no. 12 of 2012 concerning higher education. To ensure adherence to these principles, every HEI has developed their own strategic plan according to their vision and mission statements. HEIs must also synchronize their business processes and technology strategies.

Indonesia currently has 3821 higher education institutions consisting of both public and private institutions [1]. These institutions may essentially have similarities in terms of their business process and information system (IS) needs, but individually they have developed their own information system architecture. This leads to a host of issues including inconsistency in information system development, lack of a standardized data structure and lack of data integration between one information system and others. An information system in this case can either be a web information system (WIS) or a non-web information system (traditional information system). Given the substantial number of HEIs in the country, the foregoing issues need to be dealt with immediately to prevent a whole new set of problems from emerging. This calls for the need to design an IT architecture model that HEIs in Indonesia can adopt.

Higher education institutions require a proper standard and model that can be implemented to enhance alignment between business strategy and their existing information technologies. Recently, HEIs in Indonesia still have inadequate options for applying an appropriate IT architecture model. Existing systems are mostly confined to a narrow scope for specific educational institutions [2]. However developing the required model is a complex task and need a complete methodology in order to simplify the implementation. This study shall apply the Enterprise Architecture (EA) and Model Driven Architecture (MDA) concepts for formulating an IT architecture model into activities related to web information system development to meet HEI needs. Characteristic similarities in HEI business process shall serve as the basis for designing the appropriate architecture model. Stages involved in this design process are critical to lay the foundation for the subsequent phase of designing the architecture or blue print, as output for the IT architecture model. Information system development for HEI shall then be based on the IT architecture model. This IT architecture shall guide the institutions in developing an integrated information system to support HEI goals.

By applying the IT architecture model, the HEI can evaluate their existing information systems and technologies, while defining the need for a new information system and technology. Several principles and platforms planned within the IT architecture can lead to changes in the existing information systems and technologies. Concerning the information system, HEI is definitely vulnerable to changes as it relates to user needs in conducting the business process. In light of this, the information system must be adaptable to these changes. The same situation applies for the need of a new information system. Different platforms between existing information systems make it difficult for the information system to interact with each other which results in its ineffective use. Under such environments, a new information system can function as an interface between several existing information systems.

Alterations of existing information systems or the development of a new information system for university must be done within a time span acceptable to the current business condition. On the other hand, software developments to support information system modifications are still incapable of keeping pace with current needs. Application generator emerges with the promise of resolving these issues by developing a system within a relatively short space of time, along with an easier maintenance system. This study shall adopt the MDA and SOA approaches to develop a new WIS and integrate existing WISs within the institutions. The approaches helps develop a model based on how services are delivered through a WIS. The functional requirements can be investigated by identifying services from the basic elements in the existing systems. Integration processes and services from previous models can subsequently be transformed into the required models with a specific web platform.

After developing and implementing the web information system, evaluation is an important aspect necessary for gleaning information on the extent to which the objectives of a WIS has been successfully achieved. Through evaluation, feedback will also be generated, crucial for improving WIS quality in the future. The main constraint in conducting evaluation is in determining the evaluation criteria, evaluation parameter and methodology for establishing the evaluation framework. However the various traditional information systems evaluation model in the literature usually are not comprehensive focusing on some dimensions of web information system concept [3]. This thesis will be complemented with a study about new model for evaluating WIS that allows us to assess quality and effectiveness of developed WIS.

Based on the background to the foregoing issue, this thesis shall examine three key aspects – architecture, development and evaluation – related to WIS in an educational institution which shall each be described in detail in chapters presented in this thesis. Implementation is a complementary aspect that this thesis will examine in the form of a case study. All aspects examined in this study will be incorporated into a framework research named the WIS education framework. This Framework will provide the guidance when architecting, developing, implementing and evaluating WIS in educational institutions.

1.2 Problem Statement

Several methods and approaches for web application development have been adopted in previous studies, all of which are intended to develop successful, complex and quality web-based systems [4]. Nevertheless, only a few has been specifically explored for the interest of the education domain. Web information system for education is relatively distinct from other enterprises due to dissimilarities in terms of purpose, characteristics of problem and users. Higher Education Institution requires the appropriate standard and model in order to establish a WIS architecture that is both suitable and straightforward with regard to its development process and evaluation. Then, to implement the WIS architecture, an appropriate method is necessary to ease the process of WIS development, integration and evaluation. This also implies that HEI requires an appropriate approach to integrate all work related to enterprise architecture and the development of its information system. All the study in this thesis is an attempt to improve the development of WIS in educational institutions by considering above problems, which can be stated by the following three part of research questions:

1. How can HEI design an IT architecture that can be integrated into both their internal and external systems? How should an IT architecture model be formulated to ensure its compatibility with the business process and information system needs of the HEI concerned?
2. How can HEI develop and integrate existing WISs in a speedy, easy and efficient way? How should a WIS development process be defined in order to facilitate the construction of WIS from models to automatically obtain web applications?

3. How can HEI evaluate WIS that has already been developed and implemented by using the appropriate WIS evaluation model? How should the WIS evaluation model be constructed to guarantee that the evaluation results provide insight into the quality and effectiveness of the WIS being developed?

1.3 Research Objectives

The research objectives of this thesis is developed to answer the three part of research questions presented above:

1. To formulate an IT architecture model to resolve information system characteristic problems in educational institutions. This model can be used as a reference for HEI so that IT architecture can be obtained compatible with the business process and information system needs. The proposed IT architecture model based on Enterprise Architecture and Model Driven Architecture.
2. To provide WIS development method that can be used for developing and integrating existing web information systems. This method defines the WIS development process to automatically obtain web application interface from information system models based on Model Driven Architecture and Service Oriented Architecture.
3. To construct an evaluation model for evaluating WIS quality and effectiveness. This evaluation model can be applied to assess the quality and impact of WIS development and implementation in educational institutions.

Figure 1.1. shows the conceptual framework of this research.

The WIS educational system in this research framework is proposed as a structured approach to guide all processes related to WIS development in education. These processes begin from the construction of an IT architecture model to the evaluation of the WIS that has been developed and implemented. We follow a well-defined system development life cycle to ensure that WIS development is studied from different perspectives. The implementation of the developed WIS can then be evaluated by involving end users as stakeholders in educational institutions.

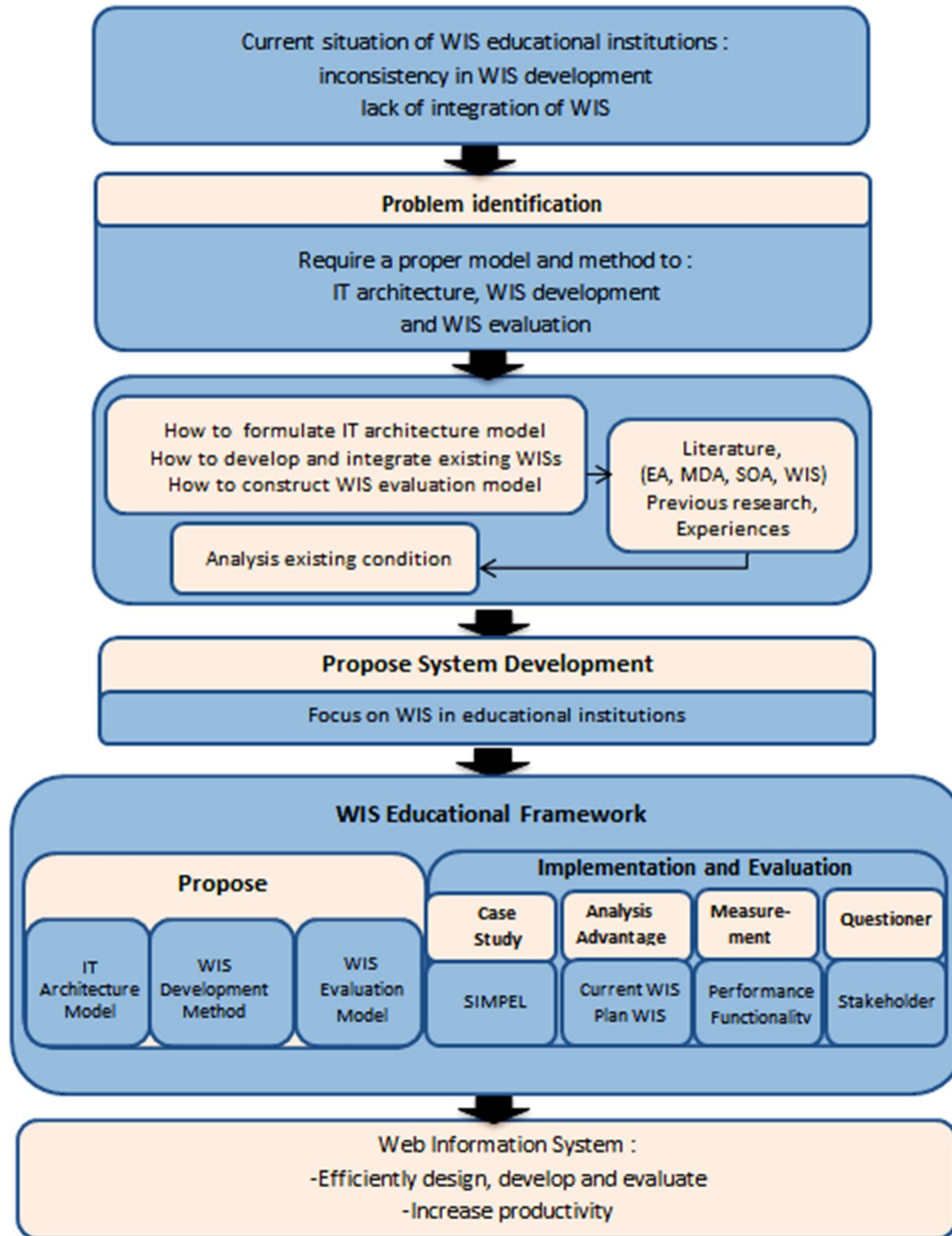


Figure 1.1 Research Framework

1.4 Thesis Contribution

Three key significant aspects – an information technology architecture model, a WIS development method, and an evaluation of WIS model – incorporated into the WIS education framework research are highlighted in this thesis in contributing towards resolving issues

related to the development and integration of existing WISs in educational institutions. This is in a way that the planned WIS can effectively meet institutional needs, and can be developed and evaluated in an easy, quick and efficient way. The result of this study is expected to specifically contribute to the advancement of educational institutions in Indonesia, particularly as reference for designing and developing a comprehensive of WIS.

1.5 Organization of Thesis

This thesis is organized into seven chapters. The outline of each chapter is as follows:

Chapter 1: Introduction

This chapter introduces the background of this thesis. It presents about a number of problems and shortcomings regarding the architecture, development, and evaluation of WIS. It also identifies integration needs of existing WISs in educational institutions. The problem statement, research objectives, thesis contribution, as well as the organization of thesis is presented in this chapter.

Chapter 2: Key Concepts and Technologies

This chapter presents the key concepts and technologies that we used in this thesis. In this state of art well defined web information system is presented with its methodologies, integration and interoperability issues, the case of HEIs in Indonesia. Model Driven Architecture (MDA), Service Oriented Architecture (SOA) and Enterprise Architecture (EA) are described as a conceptualization of IT architecture models. It also includes the related work of this thesis.

Chapter 3: EA-MDA Model to Resolve IS Characteristic Problems in Educational Institution

This chapter describes the current situation of information system in educational institutions in Indonesia, WIS development strategy and Architecture used for creating the foundation for

WIS Development. The proposed IT architecture model that provides clearly defined directions with regard to inputs and outputs for EA development activities is presented. The two approaches are mapped into an IT architecture model for enterprise architecture definition, which influences the abstraction levels of MDA and the comprehensive views of EA framework. The proposed architecture model then applied to create IT architecture for university needs by using a case study.

Chapter 4 : MDA and Integration of Web Information Systems

This chapter describes the proposed method for developing and integrating web information system based on MDA-SOA. The WIS development method comprises five main steps : creating business models, creating process models, defining templates, code generator and web application source code. The developer of WIS need guidelines to assist them in the WIS development process. The steps that can be followed in development process consists of Problem Identification, Requirements Specification, Define Data and Web Services, Design Models, Generate Source Codes, Complete and Customize the Codes, and Finalize the Process. The proposed method also provides templates to transform model to text. By using the defined templates and models, the code generator creates all application source code based on Model View Controller (MVC) pattern, Data Definition Language (DDL) and Web Service Description Language (WSDL) files.

Chapter 5 : SIMPEL : A Case Study

This chapter presents a development of WIS by applying the proposed method to integrate two existing web information systems. A case study conducted by designing the SIMPEL as an web interface between academic information systems and Course Management System at University of Lampung (Unila, Indonesia). It also describes the functionality, service categories, the generated web application of SIMPEL and completed by demonstration scenario. The

implemented SIMPEL is then analyzed in the teaching/learning activities at Department of Electrical Engineering Unila.

Chapter 6: Evaluation Model for WIS Quality and Effectiveness

This chapter proposes an evaluation model for evaluating WIS quality and effectiveness. It is generated using the fundamental theories of both the TAM and the IS-Impact models with 12 dimensions of measurements. This evaluation model is then applied to assess the quality and effectiveness of SIMPEL implementation in Unila. It is also describes measurement result consisting of the performance evaluation and functionality evaluation.

Chapter 7: Conclusion and Future Work

This chapter concludes the results of this thesis with its significance. Some future work, which cropped up during this research and has not been covered because of the time shortage, is discussed.

Chapter 2 Key Concepts and Technologies

This chapter presents the key concepts and technologies that we used in this thesis. In this state of art well defined Web Information Systems is presented with its methodologies, integration and interoperability issues, the case of HEIs in Indonesia. Model Driven Architecture (MDA), Service Oriented Architecture (SOA) and Enterprise Architecture (EA) are described as a conceptualization of IT architecture models. It also includes the related work of this thesis.

2.1 Web Information Systems (WIS)

The rapid pace of IT development has led to an increasing number of information systems being transformed into web information systems [5]. Many organizations use websites to easily distribute information to employees or customers. Comprehensive information structure and flexibility is the main reason for the use of WIS. A web information system that is intended in this paper refers to an information system that provides the facility to access web-based complex data and interactive services for users [6]. Web information system is an effective platform for collecting, storing, managing and disseminating information similar to the function of traditional information systems. The difference however is that WIS can handle vast amounts of information from diverse sources undergoing fast-paced technological advancements, in different formats with high levels of complexity. Some examples of WIS are: e-commerce, educational websites, web order tracking system, e-banking etc. The objective of the system is to provide a platform for users to reach their goals. One characteristic of this system is a data-intensive applications. The data comes from a different source or some different sources that need to be integrated. This web-based system can be integrated into

other WIS or non-WIS for enterprise purposes such as integration with database or web services.

Web information system is part of web applications that have various scope and complexity from small-scale web sites to large-scale applications [7]. Recently WIS has become a complex enterprise application. WIS normally integrates existing systems by using interactive interfaces, handles a large number of users with different access rights, and accessible through various of devices. From its development aspect, web-based systems are now more than simply about visual and user interface design. A systems developer must be familiar with the environment and current needs, and armed with the latest approaches that allow users to adapt quickly to the system. As such, WIS development is a complex task that requires approaches, strategies and effective tools to assist the developer [8].

Educational institutions have been using WIS in order to present updated and well-organized information, and to lead effective communication with their users. Efforts to migrate or develop new WIS for educational institutions are possible through self-developed solutions, in addition to commercial solutions, whereby available sources can be utilized and modified accordingly. Some existing information systems designed for business enterprise purposes are modular and open, including common functional modules intended among others for financial, resource management and marketing. However, systems for educational purposes must receive adjustments in order to respond to the specific needs of academia. Systems established within HEI normally include specific modules to support different management areas that among others deal with admissions, academic affairs, students, campus, finance and accounting, learning and grants.

2.1.1 Methodologies for Web Information Systems

Web engineering is a discipline that proposes a systematic approach to the successful development of web applications [9]. As in software engineering, web engineering applications highlight the need to design carefully prior the implementation phase. The existence of reusable components also further facilitate the development of new web applications. Web engineering uses similar concepts, methods and activities for software engineering and has adopted some of the principles of software engineering, however, there are differences in how

these methods and activities are conducted [10]. Specificity in web engineering than software engineering is to consider the peculiarities of web application needs, such as presentation issues, informational content, navigational aspects of the applications. Authors in [11] indicate the factors that distinguish web engineering from software engineering. Those factors confirm that the approach to web development should be more emphasis on the uniqueness of web applications.

- Most of the web-based system development requires a short time development. Using the similar level of planning and testing in software development will be more difficult.
- Most of the web-based system are content-driven. Developers should consider the characteristics of different media, including text, audio, video and graphic in developing of content.
- Web-based systems focus on presentation and interface, visual creativity and incorporation of multimedia. It needed emphasis on more creativity and visual presentation when presenting content to the end-user.
- Web-based system is a various delivery medium, may operate on the Internet or Intranet. Conventional software usually operates in a well-defined environment.
- Most Web-based systems are evolutionary by nature. They are evolving all the time and need to be developed within a short time.
- In the development of Web applications, developers should consider about variety of skills of users, and the different characteristics and preferences. Thus the user interface design and personalization of information should actually consider these aspects.

Web information system design is an extremely complex task. When designing a WIS developer should consider all the features of WIS such as data integration, automatic generation of a presentation, personalization etc. That requires a methodology that can drive every phase of development and help reduce the development effort of WIS. Numerous design methodologies have been proposed to assist developers to develop WIS such as model-driven methodology. This methodology uses the model to determine the different aspects of a WIS. Model-based approach for WIS design has many advantages such as a better understanding among stakeholders of the system, increasing system maintenance, the possibility to check the

validity and consistency between models, etc. A range of available web engineering approaches, such as Web Modeling Language (WebML) [12], Object Oriented Hypermedia Design Model (OOHDM) [13], Web Site Design Method (WSDM) [14] or UML-based Web Engineering (UWE) [15] also provide excellent methodologies and tools for the design and development of web applications.

To develop different types of web applications on different models and phases, diverse web application development methods are available, such Unified Process, Entity-Relationship (ER) and Unified Modeling Language (UML) [16]. The UML is the most widely used for modeling web applications architecture includes a client browser, a web server, and a connecting network [17]. UML defines a visual language that is used to model a particular domain or system. Web applications are develop on different sets of web components, including traditional and non-traditional software written in different languages and frameworks such as Active Server Pages (ASP), PHP: Hypertext Preprocessor (PHP), Java Server Pages (JSP) and Asynchronous JavaScript and the extensible Mark-up Language (AJAX).

Web applications development is an activity that has a short lifetimes. It required the appropriate method and tools to enable the system to provide continuously updated information. Model Driven Architecture [18] has been widely proposed for simplification and automation with concern to the development of web applications. It upgrades the quality and speed of web application development, as well as its maintenance. One of MDA's strong points is the ability to generate automated tools and services to describe the abstraction models and also facilitate the transformation process between models. It is also important to define templates, as model transformations that will transform a given model to another model/text and to generate the target system [19].

2.1.2 Integration and Interoperability Issues of Web Information Systems

Integration and interoperability are important issues in WIS. To ensure that data and information is effectively shared, the diverse web components need to interact with each other. There are several benefits of developing integrated web applications:

- Reduce manual processes and repeated activities in generating the same information

- Control the major activities
- Collect, store and distribute information quickly
- Increase data consistent, reliability, integrity and validity
- Integrate useful services into new functionality
- Introduce new procedures that will eliminate existing inefficiencies.
- Ensure the availability of information needed for decision making
- More flexible to users to access data for reporting

In term of integrating new information system with the heterogeneous existing systems in institutions, author in [20] proposes three solutions, namely: (1) All-in-one solution. (2) Self-evolving and light level integration. (3) Heavy-level integration. According to the author the first solution has these characteristics: highly cost of development and user training, greatly risk, long schedule, long term benefit to business is efficient or good, and it also provides real-time response. However considering the reasons given in [21], some HEI decided to provide an all-in-one solution to develop a completely new system solution in their institutions. It seems to be the best option since it can integrate various existing web information systems and applications into one interoperable system and relatively easy for future modification and revision.

Existing web information systems can be integrated in various ways. A common integration approach is through the use of web services where a separate interface can also be used by other web applications. EXtensible Markup Language (XML) is one of the technologies applied by web services to support data exchange by combining it with business logic, and later accessed via the internet [22]. Through the use of web services, data duplication among work divisions can be avoided and data inconsistencies effectively eliminated. Furthermore, data ownership from work divisions is assured and systems security heightened as users are denied direct access to data sources.

Interoperability refers to the ability of a system to communicate, exchange information, and cooperate with other systems [23]. Interoperability is the most important principle of Service-Oriented Architecture (SOA) that can be realized through the use of web services. In SOA, a problem will be divided into several cooperated small services. For users of services, is not

important to know how the service is implemented, how it is stored and how they are structured. Users only need services so that the system can occur between data synchronization and data exchange. Web service technology on the other hand allows for interoperability between web applications. By application interoperability is meant not only the WIS interoperation at data level (the output of one system is the input of another system) but also the ability to reuse existing WISs models in developing new WIS. Web services currently available for use by systems have grown exponentially. As a consequence, there has been rising demand for the integration of web services into web application development.

2.1.3 WIS : The Case of Higher Education Institutions Indonesia

Each higher education institution has its own set of strategies built on the stated vision and mission. Information system must therefore be well planned to complement this strategic direction. Pursuant to government regulations in Indonesia the information system established by university must be prepared to support academic program management and quality improvement. The information system must help university meet its Quality Assurance (QA) standard, that should at least consists of data collection, analysis, storage and retrieval, data and information presentation, and communication with relevant parties [24]. The information system evaluation standard in HEI is regulated by the government and covers several key elements, including the availability of the following components [25]:

- blue print on information system development, management and utilization, including systems that regulate data flow, data access authorization and disaster recovery systems.
- decision support systems to support top management for better planning and self-assessment analysis and more objective decision-making.
- database that at least should consists of information on finances, assets, facilities and infrastructure, academic administration, student and alumni profile, and teaching staff and supporting personnel.
- information system intended for campus internal and external communication, and access to sources of academic information for students and teaching staff.
- internet capacity with adequate bandwidth ratio per student.

From the aforementioned elements, information systems that supports HEI activities at a minimum should encompass academic, human resource, financial and asset information systems that ideally web-based platform. An integrated WIS is crucial for the academic community to ensure easy access to data and information for learning, administrative and reporting purposes. Management also requires an integrated WIS to support the decision-making process and for monitoring and evaluating HEI performance. Given the purpose of a HEI, and its WIS requirements along with the business process involved, it can be concluded that the data and information characteristics of a HEI are as follows:

- Distributed in every working unit and typically have different structures and standards
- Accessed by all members of the academic community with different needs, roles and level of knowledge
- Data is obtained from historical and operational data of the HEI
- Continued rapid data growth
- Immense data volume as academic data must be stored for a lengthy period
- Data must be periodically transferred to a different system as it has become input for another system
- Data transactions with varying time periods depending on academic calendar, peak time for certain periods such as the new academic year
- For data reporting purposes to the central government, academic data must be based on data structure and relations set forth in the National Higher Education Database Data Glossary determined by the Directorate General of Higher Education (DGHE) ²[26].

2.2 Model Driven Architecture (MDA) and Service Oriented Architecture (SOA)

Designing and development of WIS in educational institutions is a complex task. This type of work involves many complex business processes and various different technologies. The design process becomes more complex with the use of these technologies. To produce an effective

² DGHE stands for Directorate General of Higher Education. This is a government agency (under the Ministry of National Education of Indonesia) that oversees Indonesian higher education.

approach for the purpose of improving productivity and interoperability, the Object Management Group (OMG) develop MDA as a reference to achieve wide integration of enterprise models and software applications. In term of integration, recently both MDA and SOA have emerged as a major evolutionary stage in enterprise integration areas [27].

Model transformation in MDA is defined as a mapping to define correspondences between source and target models, while SOA provides an architectural platform and the required concepts that allow for simplifying integration. By combining MDA and SOA, two core advantages will be generated from each approach. The models and information based on service derived from the SOA usage as well as use transformation and automation made possible from the MDA. Consequentially, systems integration can arise at different abstraction levels and transformed to low-level forms. Certain systems have demonstrated the ability to successfully combine SOA and MDA. We briefly summarize MDA and SOA concepts and how they can be combined.

2.2.1 Model Driven Architecture (MDA)

The MDA comprises of three fundamental phases that are analysis, design and model-driven implementation. Its overall features correspond with :

- **Computation Independent Model (CIM).** A computation independent model is a view of a system from the computation independent perspective. A CIM does not show specifics of the structure of systems. A CIM is called a domain model occasionally and a terminology that is familiar to practitioners of the domain in demand is used in its specification.
- **Platform Independent Model (PIM).** A platform independent model is a view of a system from the platform independent perspective. A PIM shows a specified degree of platform independence so as to be appropriate for use with a number of different platforms of the similar type.
- **Platform Specific Model (PSM).** A platform specific model is a view of a system from the platform specific perspective. A PSM combines the specifications in the PIM with the details that specify how that system uses a particular type of platform.

Another characteristic in MDA is that it enables different applications to be integrated by explicitly relating their models; this facilitates integration and interoperability and supports system evolution (deployment choices) as platform technologies change. The MDA approach in the software development lifecycle can be adapted to any type of technology in different domains. The MDA process extends from the core implementation of models to the specific Implementations using different technologies, programming languages and platforms. The core of MDA includes widely-used OMG modeling standards: UML, Meta Object Facility (MOF) [28] and Common Warehouse Metamodel (CWM) [29]. The UML as a standard notation for constructing models of object-oriented software allows an application model to be constructed, viewed, developed, and manipulated. In MDA, UML is used for visualizing, storing, and exchanging software design and models. MOF is a model-driven framework for specifying, constructing, managing interchange and integrating metadata in software systems. It represents metamodels and how to manipulate them. CWM is a specification that describes metadata interchange among data warehousing, business intelligence, knowledge management and portal technologies.

MDA provides an architecture that assures :

- Platform Independence: allowing a reduction in time, cost and complexity of developing applications with different platforms
- Interoperability: using methods to assurance that standards based on multiple implementation technologies all implement identical business functions. Interoperability in the MDA is when PSM are targeted at different platforms, they can interact each other due to elements from one PSM have well transformed into elements in another PSM. MDA addresses this problem by generating not only the PSM, but the necessary bridges between them. As an example in the case of a PIM have transformed into several PSM targeted, all the information needed to bridge the gap between all of the PSM is available.
- Productivity: allowing developers, designers and system administrators to use a language and concepts they are comfortable with, while allowing seamless communication and integration across the teams.
- Portability: increasing application reuse and reducing the cost and complexity of application development and management. Within the MDA, portability is realized by

focusing on the development of CIM and PIM, which are platform independent. A PIM can be automatically transformed into several PSMs for different platforms. All that is specified at the CIM or PIM level is absolutely transportable.

2.2.1.1 Metamodel and Model

Metamodels simplifies the communication about models. It can be seen as the model whose instances are the types in other models or as mapping of meta-models elements to the modeling language's elements. A well-known metamodel is the specification for UML, which captures the classes in a developer's model. In MDA, metamodel facilitates the mapping and transformation between models [30]. Models are the primary artifact for the developers to work on which is different with the traditional technique where the programming language is the primary concern. A significant factor in the development of web application system is the support of models – using a common language – to avoid misunderstandings. In this case the UML is appropriate with these requirements. The UML notations and diagrams can be used to describe software systems at a high level of abstraction. Models are created by using the UML diagram and then part of the code is generated automatically using a code generation tool. The rest of the development is done manually by writing custom code. Consequently, there is high dependency of code on model. Even a slightest of the change at system modification time can create significant impact on the already generated code.

The MDA approach which discussed in this work helps develop a model based on how services are delivered through a web application system. The functional requirements can be described by identifying services from the basic elements in the existing systems. Integration processes and services from previous models can subsequently be transformed into the required models with a specific platform. One of the advantages of the MDA approach with regard to web information systems integration is that it is capable of separating different models at varied levels. Separate and different levels of models will increase flexibility in handling the complexity of the process. In line with technological changes, there will be no need for changes to CIM and PIM. On the contrary, it would be necessary to make adjustments to PSM as it would be simpler and less costly than having to change the overall software.

2.2.1.2 Transformation and Code generator

The concepts in MDA separate the application modeling away from the specifics implementation. The developers allow to create a model without knowledge of the other models in the system. The existing models can be combined in the final step to create a whole system. The application is independent of its implementation and can be recombined with other technologies in the future [31]. MDA concepts are handier to the problem domain than those proposed by programming languages. However, the advantages of MDA are based on the assumption of the transformation issue. Some research discussed about the transformation application among models such in [32].

The authors classified their approaches into three categories:

- a. Direct model manipulation : the transformation logics are contained that can access source models and transform those into codes.
- b. Intermediate representation : the model transformations are performed through a standard intermediate form.
- c. Transformation language support : it uses a specific language to specify the model transformation.

Transformations can be divided into two main categories namely Model-to-Model and Model-to-Code Transformation [33]. Model to model transformations are used to bridge large abstraction gaps between PIM and PSM through generating intermediate models which are more close to PSM. On the other hand, Model to text transformations are used to generate program code from models. The model transformation languages that currently most widely adopted, such as ATL (ATLAS Transformation Language) [34] for specifying Model to model and Aceleo [35] for Model to text transformations supported by a set of development tools built on top of the Eclipse environment [36]. Model to text transformation is one of the essential steps in the MDA, which makes it possible to generate an extensive amount of code from models. This automation can reduce development time and improve code quality. In this thesis we use Aceleo as an innovative MDA code generator built upon all OMG concept. It is categorically customizable and can be used for PIM-to-code or PSM-to-code generation. In Aceleo transformation rules are written in templates that produce text-files after execution. The template is called mtl file.

2.2.2 SOA Technologies and Web Service

SOA is modelled as a relationship of three kinds of participants : the service consumer (client), the service registry, and the service provider. The service provider publishing its existence in some registry, followed by the service consumer finding the service provider by querying the service registry. The service consumer then binds directly to the service provider. This interprets to the publish–find–bind architecture pattern. By using SOA, applications can interoperate each other in the broadest set of service providers, and providers can effortlessly be changed on-the-fly without any modification of application code. Certain technologies adopted to implement SOA are Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI) [37]. All of these are main techniques for data exchange, data integration and data sharing between existing systems.

Recently, SOA has increased more popularity mostly due to web services. Web service are becoming gradually important for integrating systems and services. In this thesis we employ web services as they are one way of implementing SOA, with some advantages. To implement SOA it is not necessarily using the web service, but it is recommended procedure due to a service in SOA can be represented well by using the web services. The work environment of web service is shown in Figure 2.1. Some characteristics of web services such as :

- An application logic that can be accessed and published using standard Internet (TCP / IP, HTTP, SMTP, FTP, JMS, Web).
- Described in XML format.
- Identified by Universal Resources Identifiers (URI)
- Loosely coupled, self-contained, modular and open (non-proprietary)
- Used to support interaction interoperability of machine-to-machine network via the Internet / Intranet.

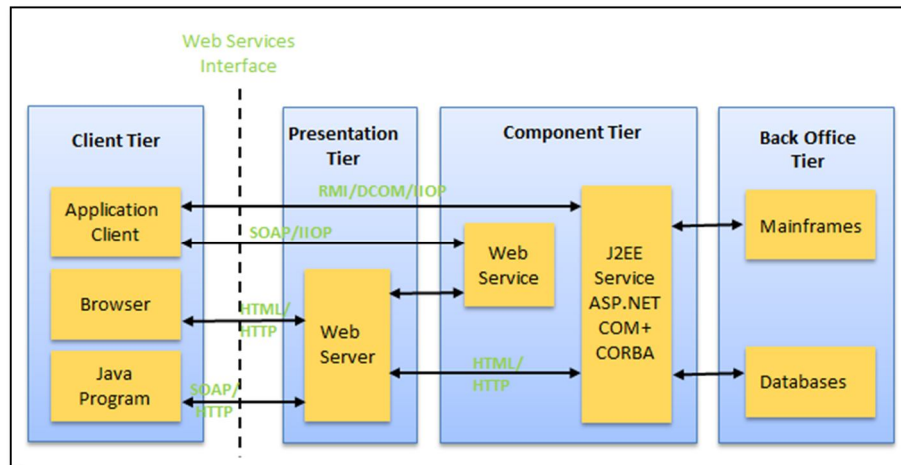


Figure 2.1 Work environment of web service

Web services are functional components, described in the WSDL [38]. Many web services are now available and it therefore seems accepted to reuse existing web services. In model-driven development the models can be used to describe such as structure, user requirements, activities. The models of web services seem to be useful. In the case of web service development the models are transformed into the WSDL. The usage UML diagrams for automatic generation of WSDL service descriptions is described in [39]. This research discussed about WSDL-dependent UML profiles which is about a UML profile for WSDL and introducing WSDL-dependent stereotypes. According to this profile, a web service should be modeled using the specified UML stereotypes and tagged values, resulting in a WSDL-dependent model. However, author in [40] state the advantages of WSDL-independent UML models in order to enhance the understanding and efficiency of service. They described a model-driven web service development process, where web service descriptions are imported into UML models. Then they presented a to-ways mapping between a WSDL-independent service model in UML and the corresponding service description in WSDL. We applied this concept as they state WSDL-independent UML models simplify building of web services, especially when the services are complex.

Generally, in order to produce a feasible web application which able to integrate existing information systems, the following key questions need to be answered: how to align the new system with running systems and strategic plan of the university? What basic services are

included in the integration platform? How can web application modeling be viewed from different levels? How to adopt web services and software reusability? What methods are used to abstract the complexity of business processes? How to utilize available technology and architecture for developing platform-independent models to generate code automatically?. This work will specifically provide answers to the foregoing questions in a systematic manner which will then become the basis for developing the web information system in educational institutions.

2.3 Conceptualization of Information Technology Architecture Model

Compatibility between information system application and institutional needs can be assured by taking into account the integration factor when developing the system. The essential purpose of integration is to narrow gaps found in the systems development process. Enterprise architecture helps reduce these gaps by providing model-based information system planning, designing and managing within the enterprise. With regard to HEI, the EA concept is highly relevant to be applied for maximizing the benefits of having an information system in the educational institution. The information system can therefore strategically increase HEI comparative advantage.

2.3.1 Enterprise Architecture (EA)

Enterprise is perceived as a complex system consisting of diverse and interdependent components. It is bordered by an external environment which impacts the enterprise operations. It also provides the several inputs that are transformed by the enterprise components to construct the output in the form of products and services that are returned to the external environment [41]. Enterprise Architecture is a framework that defines the relationship between the enterprise components and outlines how an enterprise achieves the existing and upcoming business objectives.

An Enterprise Architecture is frequently used to help with the understanding of the structure and functioning of the systems, and the roles and expectations of many stakeholders in a complex enterprise such as developing a large information system. Enterprise Architecture in general has the following domains: Business Architecture, Information System Architecture,

Technology Architecture. Each domains is a separate architectural discipline and EA integrates each of these disciplines into a cohesive framework as shown in Figure 2.2. Business Architecture defines the business strategy, management, organization and business process. Information System Architecture comprises Data Architecture that describes data structure, data management and resources; and Application Architecture that defines applications required for managing data and supporting business functions. Technology Architecture represents software application infrastructure to support application development.

Enterprise Architecture has various frameworks and those that are often applied include Zachman Enterprise Architecture Framework [42] and The Open Group Architecture Framework (TOGAF) [43]. These frameworks essentially serve the same purpose of facilitating the design of the IT architecture of an enterprise. Nevertheless, preference over a particular framework also needs to be consistent with enterprise needs [44]. Those frameworks are systematic models that organizes descriptive representations without describing an implementation process and they are independent of specific methodologies.

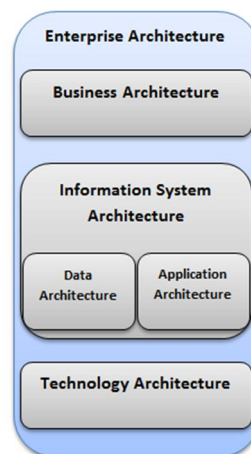


Figure 2.2 Enterprise Architectural Relationships

There are a number of valuable concepts that are significant and reflect the general architectural description of an enterprise. These concepts are necessary to comprehend the information system development. The key concepts and relationships among them are described in Figure 2.3 [45].

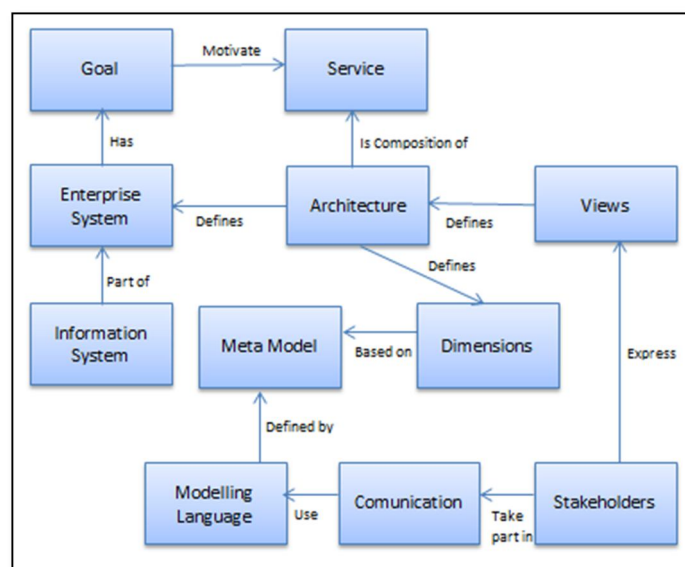


Figure 2.3 Modified version of EA concepts and relationships

The illustration of this concept in this figure is as follows : An information system is embedded in an enterprise system. Autonomous services are motivated by different enterprise system goals. Architecture defines the enterprise system, which is expressed by the views of different stakeholders and can be characterized in different dimensions. Behind every graphical representation lies the description of the stakeholders' view that together provide the integrated architectural description of the enterprise system. Dimensions are based on different models that are defined by various modelling languages. Metamodels define the modelling notation of graphical language. The modelling language is necessary for an unambiguous understanding of diagrams and for communication among stakeholders. The purpose of communication is to bridge the gap using shared models and methods with the purpose of developing the comprehensible descriptions of EA.

Various models or a model collection are used to represent these aspects. The MDA, as described in earlier, is an approach to create models, and generate code from models. It is a generic approach that can be used with any existing methodologies including EA Framework. One of objectives of this work is to establish the importance of EA and its design principles and it's mapping with MDA.

2.3.2 Enterprise Architecture and related MDA Concepts

A key factor of EA is the compatibility between business and technology available for all stakeholders. This can be attained when all involved parties share a common perception. In this regard, modeling becomes a crucial aspect that needs to be taken into account [46]. Several models being applied however are still difficult to understand and have backgrounds that are not commonly used. Under such circumstances, the advantages of the preferred framework may not be optimally realized. A model refers to systems specification and normally presented through illustrations and texts. The ability to develop and transform models at different levels is a critical feature of MDA as mentioned in earlier section.

The TOGAF Architecture Development Method [ADM] [47] and MDA are mutually complementary and present immense business potential if effectively applied in combination. Hence, synergies between TOGAF and MDA will lead to improvements for the organization particularly for developing better architecture quality. From 9 phases of TOGAF ADM and requirement managements phase, the 6 specific phases have applicable and useful criteria to map to MDA. The applicable criteria are phase A (Architecture Vision) to CIM and PIM levels of MDA, phase B (Business Architecture) to CIM and PIM levels, phase C (Information System Architectures) to PIM level, phase D (Technology Architecture) to PSM level. The useful criteria are phase E (Opportunities and Solutions) and phase G (Implementation Governance) to PSM level. The proposed model in this study based on this approaches [48].

The solution to apply MDA in sync with other frameworks in order to establish the required model is expected to address any existing modeling issues. The model-driven method is a systems development approach in which the model is described in a way that clearly defines system comprehension, design, development, application, maintenance and modification. MDA in this case is intended to develop an information system that can be applied to describe enterprise business and resources. Through this form, the enterprise will gain the ability to generate specific applications and if required, make the necessary modifications according to changing needs which will later be represented in code form. Similarly, using SOA will complement the WIS being developed. Web services, as the applied technology, on the other hand shall regulate on how the system will interact and be utilized by WIS or other applications.

To effectively implement SOA, it is therefore crucial to ensure an accurate analysis of the required data, ongoing business process, as well as applications and interfaces employed in running the business process in every unit within HEI. The simultaneous use of all of these approaches shall lead to a well-defined IT architecture understandable to all stakeholders, while MDA-SOA helps ensure a more effective and manageable systems development process.

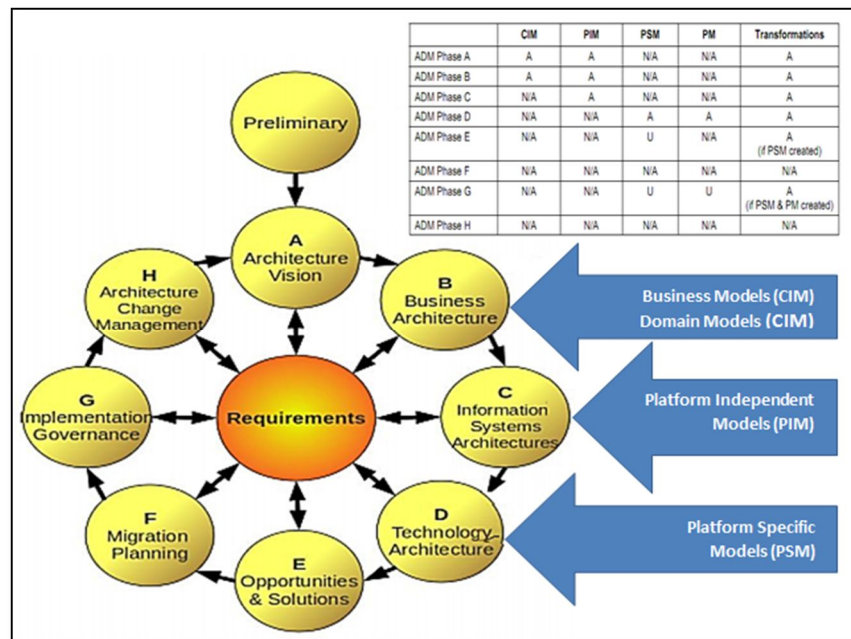


Figure 2.4 TOGAF ADM and related MDA concepts

2.4 Related Work

Integration between business and information technology is a major challenge that industries, including HEIs must deal with. The adoption of the Enterprise Architecture concept in this case becomes an absolute need [49]. Enterprise Architecture is vital in information systems development as it provides insight into the organisation's structure, business processes and technology that support the business and construction the whole enterprise [50]. In the context of large information system development, an EA is often applied to support with the understanding of the structure and functioning of the systems. Authors in [51] have presented an architecture framework based on EA that can serve as a strategic guide to the development of web systems.

Various models in different abstraction levels required by EA can be supported by MDA [52]. Applying the EA and MDA together is the key element for optimizing the business processes of an enterprise. Enterprise Architecture provides a comprehensive understanding of the core business process of an educational institution and defines the information system that contributes to the optimizing of the business process. Enterprise Architecture essentially focuses on strategy and integration. MDA makes use of models as its main element while focusing on efficiency and quality. The author in [53] has described that the MDA approach can support modeling at hierarchical systems employed for aligning business and IT within an organization. The research also featured techniques applied to design hierarchical systems and developed an integrated enterprise model. A number of studies have adopted the MDA and EA frameworks to tackle modeling issues, such as [54] that described the conceptual mapping between the MDA and Zachman framework and [48] described mapping between MDA and TOGAF framework. However, research which is applying the advantages and merging the two methods to create the comprehensive enterprise architecture model remain few. The model required for guiding important activities in the development of enterprise architecture in University. We formulate an IT architecture model that provides clearly defined directions with regard to inputs and outputs for EA development activities. Enterprise Architecture captures a higher educational institution current (as-is) and planned (to-be) capabilities using a number of models and required to develop the models further. The MDA drives the models that provided from the previous activities.

MDA on the other hand has been widely proposed for simplification and automation with concern to the development of web applications. It upgrades the quality and speed of web application development, as well as its maintenance. Earlier research have adopted the model-driven approach for the development of web applications. The author in [55] for example describes on how MDA is applied to support the web development process by defining models and metamodels. Specifically for WIS development, author in [56] that presented the MIDAS framework. These studies focused on the structural dimension of WIS. Still adopting this framework, authors [57] have complemented the issue on navigation model construction from a user service-oriented perspective. In order to support the whole methodology, an MDA tool (M2DAT—MIDAS MDA Tool) is being developed, which integrates all the techniques proposed

in MIDAS for the semiautomatic generation of WIS and whose early functionalities. In this work, we also provided a systematic procedure for developing web information systems. We defined web development process including CIM modeling, PIM modeling, and PIM to PSM and to web source code transformations. The templates to automate the transformation among PIM to code also provided and useful for generating web application.

In addition to the functional and navigational requirements of WIS, research that discussed on architectural features resulting in web specifications includes [58]. The authors have described WebSA (Web Software Architecture) based on the standard MDA. WebSA provides a set of architectural models and transformation models to specify a web application. By using webSA, in [59] UML profiles usage was introduced. It provides a set of web-specific models for each process right through the implementation phase. Certain approaches also adopt the MVC (Model View Controller) architecture pattern for PIM such as in [60] where the authors presented WebML and in [61] which the authors proposed the UML-MVC logical model as an intermediate system for use between UWA (Ubiquitous Web Application) conceptual design and the implementation phase. MVC as an architecture for the PIM promises the accessibility of a varied range of open-source and commercial technology frameworks to choose the different platforms, such as PHP, J2EE, .Net.

We have also adopted a similar approach, the MVC as an architectural pattern for PIM. However, a distinguishing factor in our research is the emphasis on the integration among web applications. Research on integrating WIS by defining information system models and templates for specific purposes such as for educational purposes, remains few. Most methods applied in available research do indeed allow for the automatic generation of web applications based on high-level descriptions of data and application logic. Developers will typically customize the source code produced from the generation process. Nevertheless, the customization process can immediately cease to exist if regenerated codes are established due to changes made in the data or desired model. This can further lead to the problem that the developer must change the manually written classes that extend generated classes. The lack of support for evolutionary web development from information models to obtain ready-to-use web applications (with few handwritten codes) is a major outstanding research problem.

Combining SOA and MDA means obtain two advantages from each approach. The models and information based on service derived from the SOA usage as well as use the benefits of the MDA such as transformation and automation. System integration could be performed in different abstraction levels and transformed to low level form. Some approaches have been demonstrated on how they can take advantage of MDA combined with the SOA approach. They presented the examples approach with some relevant solutions to modeling and designing the web services. For instance in [62], the authors introduced an approach to modelling and designing the complex distributed systems. The proposed method in [ten] focuses on an integrated approach to combine both top-down (requirements to software components) and bottom-up (existing tool assets). Another research in [63], the authors integrated the MDA and SOA component development to enable the development and usage of components and designed MDA and SOA to web services (MS2Web) frameworks. However, little research discusses the method for the service-oriented development of information systems. The authors in [64] have initiatives to follow this approach. They have described Service-Oriented Development Method (SOD-M) that consists of models and the rules for mapping from the business view to the information view. SOD-M distinguishes between business modeling and information system in which the information system will be developed. Business view focuses on the requirement of the business, while information system focuses on the functionalities and processes. The method includes the models and the rules for mapping from the business view (CIM level) to the information view (PIM level). The description of the PSM level discussed in [65][66] which also discuss about the web service interface model and the extended service composition model. In [67] the authors integrated solution of SOA and MDA to propose a standard-based extendable platform in network environment by stimulate the reusability and improve the interoperability between the computer systems and applications in industry. With the view of design and modeling aspects to support generative software production, authors in [68] presented a modeling of distributed systems to apply an e-business application.

In terms of generated web pages, most existing approaches focus on CRUD (Create, Read, Update and Delete)-based approach to enable users to update database, yet is insufficiently supported for generating from web services to satisfy the service-oriented view on web

application. In SOA environment the business services are not CRUD– based. Generating the application front end of web application from a database model is not appropriate in SOA, as it is common that a front end is built on existing services. However, in the lower parts of a service we still need CRUD functionality. Web application interface development utilize web services from the existing information systems nonetheless need CRUD functionality for its own database. We cover the shortcoming by describing our target model and template with two bases; CRUD and web services.

Some previous research in web application development have adopted MDA and SOA concepts. Nevertheless, the simultaneous adoption of the MDA, SOA and enterprise architecture concepts for producing standards, models and approaches which can be used to design, develop and integrate existing information system for educational institutions is remain few. Achieving integration is challenging due to differences in architectural modelling methods and model [69]. This study shall use the concepts to create an IT architecture model and also specifically emphasize on WIS development especially for educational institutions to ensure that WIS has a coherent planning, implementation and control mechanism in place consistent with the business strategy of the educational institutions.

Chapter 3 EA-MDA Model to Resolve IS Characteristic Problems in Educational Institutions

A combination of the EA and MDA concepts can be one of the solutions to overcome the complexity of building a specific IT architecture model for higher education institutions. The purpose of this chapter is to formulate an IT architecture model that provides clearly defined directions with regard to inputs and outputs for EA development activities. The two approaches are mapped into an IT architecture model for enterprise architecture definition, which influences the abstraction levels of MDA and the comprehensive views of EA framework. The proposed architecture model then applied to create IT architecture for university needs by using a case study.

3.1 Chapter Introduction

Every Higher education institution has developed their own strategic plan according to their vision and mission statements. HEIs must also synchronize their business and technology strategies to ensure adherence to Triple Principles of Higher Education Institutions. All of these can be attained by applying the EA approach within the institutions. Through the optimal performance of business processes, EA can facilitate strategic planning, align business and IT resources, and regulate the lifecycle of an information system development process [70].

Information system development is an aspect that should be part of an institution in order to support business activities and provide services to stakeholders, mainly in relation to data, information, technology and application. Information system development should be well

planned, centralized or distributed in related working units, and integrated into other existing systems. Information system has helped automate many phases in the business process previously done manually in an educational institution. In line with shifts in user needs, changes to the business process are unavoidable. If an information system fails to accommodate these changing trends, its utilization will therefore be less than optimal. Hence, such changes need to be taken into account when developing an information system. Generally, HEIs in Indonesia may have similarities in terms of their business process and information system needs, but individually they have developed their own information system architecture. This leads to a host of issues including inconsistency in information system development, lack of a standardized data structure and lack of data integration between one information system and the other.

Higher education institutions require a proper standard and model that can be implemented to enhance alignment between business strategy and their existing information technologies. Recently, HEIs in Indonesia still have inadequate options for applying an appropriate information technology architecture model. Existing systems are mostly confined to a narrow scope for specific educational institutions. However developing the required model is a complex task and need a complete methodology in order to simplify the implementation. This study shall apply the EA and MDA concepts for formulating an IT architecture model into activities related to WIS development to meet HEI needs. Characteristic similarities in HEI business process shall serve as the basis for designing the appropriate architecture model.

Applying the EA and MDA together is the key element for optimizing the business processes of an enterprise. EA provides a comprehensive understanding of the core business process of an educational institution and defines the information system that contributes to the optimizing of the business process. EA essentially focuses on strategy and integration. MDA makes use of models as its main element while focusing on efficiency and quality. MDA in this study is used to formulate the enterprise architecture model into the development activities for the IS university needs. MDA is positioned in the TOGAF ADM at the phase B, phase C, phase D, and may also in phase E and phase G, consequently MDA can be used as a detailed reference in designing the blueprint of IS due to similarity of modelling and tools used [71].

The purpose of this study is to formulate an IT architecture model that provides clearly defined directions with regard to inputs and outputs for EA development activities. EA captures a HEI current (as-is) and planned (to-be) capabilities using a number of models and required to develop the models further. The MDA drives the models that provided from the previous activities. This proposed model specifically emphasizes on information system development by applying MDA-SOA to ensure that information system help align the direction of the planning, implementation and control process to remain consistent with the enterprise business strategy. The critical task of this development process are selecting the appropriate model for each phase of EA at the right level of MDA in detail. The information systems will therefore meets the QA standard in order to improve the educational quality of HEIs in Indonesia [24]. This IT architecture model will then be applied through a case study at University of Lampung (Unila) Indonesia.

3.2 The Current Situation of IS in Educational Institutions in Indonesia

In the academic world, information serves as one of the most valuable resources that need to be well managed in order to accomplish goals set by the educational institutions. Education-related information as vital resource for the educational institutions includes content and curriculum, learning process, facilities and infrastructure, and human resource. This indispensable resource should ideally complement and support the existing business processes. In reality however, HEIs in Indonesia needs to anticipate and deal with several of issues in information systems. The following persistent issues in developing their respective information systems:

a. Integration

For this particular issue, each working unit has the same set of data yet not linked to each other due to inconsistencies in quantity and quality. The database and systems developed based on data held by the respective working unit is not synchronized with other working units nor is it linked to a centralized data storage system. This situation hampers the simultaneous process of data validation and data usage. This also leads to redundancy and inconsistency in information system.

b. Availability and easy access

Information system should be able to guarantee the availability of services whenever required. The system must function well at the time of need, particularly for critical systems. Easy access for users means the ability to access services from any place and at any time. A technology platform with the appropriate hardware and software infrastructure is an absolute requirement for implementing an information system with adequate accessibility capacity. Nevertheless, financial constraints often impede the development of IT infrastructure in HEI, thus the inability to guarantee the availability and accessibility of services required by users.

c. Security

An information system security system that cannot be guaranteed shall result in lack of user trust when using the system. Administrators must take this into account and anticipate the likelihood of security risks, both with regard to the organization's internal and external factors. They must consider the security of information stored, processed or transferred by the system and aspects related to information security include confidentiality, validity, backup and recovery. In term of information system security in HEI, equally important to consider is the ethical behavior and morality of human resources in charge of controlling the information system as they must demonstrate unquestionable integrity, honesty and trustworthiness.

d. Scalability

Scalability refers to the ability to expand further in line with increasing resources. A large organization deals with data transactions with considerable volume that tends to expand even further. Information system development must be able to anticipate changes in systems capacity and function required to handle both anticipated and unanticipated workload. This is often a problem that must be considered in HEI information system development. The changing capacity and function is due to several factors, including increased number of users, additional functions or as consequences of a specific incident.

e. Resistance

Apart from issues related to hardware, software and infrastructure, in HEI there are also issues pertaining to human resources. Users in a working unit or department may

at times show resistance in using a newly introduced system as they have been accustomed to the previous system which they consider as already effective. Resistance may also arise due to lack of knowledge and skills of human resources unable to keep pace with technological advancements in carrying out their work. Regular evaluation therefore needs to be conducted in addition to efforts in building user capacity to ensure acceptance upon the introduction of a more effective system.

3.3 Issues and Challenges

3.3.1 Integration

Higher education institutions needs to be encouraged to develop an WIS integrated into a model that is built according the needs of working units within the respective institutions or among university. Certain data required by departments and working units as inputs is essentially the same except for differences in its usage and reporting mechanism according to their respective business process. An integrated system shall significantly support interrelated business processes and optimize its use. For example, an effective academic system that takes into account the number of students will be capable of predicting the number of required human resource (in relation to human resource system), incoming and outgoing costs (in relation to finance system), as well as the number of required classes and rooms (in relation to assets system).

To ensure integration with existing systems, new technologies should be combined with existing technologies with the least possible effort without having to redevelop ongoing systems. Based on the current situation of the existing information system in university, integration may come in the following forms:

- a. Data integration of different business processes, different information system functions and processes yet related to the overall work cycle, and database integration.
- b. Availability of an interface that can form connections among existing information systems. Users need not login for every WIS in order to access the required data and information. They only need to login once to enable access to all relevant WIS. Data

that serves as output for a certain process in an information system can be the input for another information system.

3.3.2 Interoperability

Web information system development must consider the interoperability of systems. Interoperability refers to the ability of a system to work in sync with another system that allows information exchange and the ability to use the shared information. With regard to WIS implementation, the technical and regulatory aspect of data exchange needs to be taken into account. A key factor related to the technical aspect is high-level interoperability to ensure that data transfers from source to target can be done regardless of the diversity in hardware and software platforms. An open architecture solution is therefore necessary to allow the smooth exchange of data and information from a different system that works in sync with systems inside and outside of university. Apart from technical issues, interoperability also needs to be regulated through the appropriate policies. This is essential in order to ensure uniformity in format and data exchange mechanisms among HEIs for the purpose of guaranteeing high-level interoperability.

3.4 WIS Development Strategy

Given the importance of WIS development, HEI as an implementing enterprise must draw up clearly defined guidelines and plans for developing a comprehensive WIS within its organization. Based on an analysis of the overall situation of the institutions along with its general information system characteristics, the approach required to establish WIS within HEIs must have the following qualities :

- a. Able to integrate various existing standards and platforms
- b. Can be easily adapted to changes
- c. Facilitates the developer in the WIS development process
- d. Can be easily implemented
- e. Must have clear guidelines for evaluating WIS implementation

Many existing information systems have been established in the HEI but are not integrated with the exist of the systems. The strategy to retain existing information systems still in

operation, re-engineer or replace with new systems can be seen from the system modularity parameter, on whether the existing information systems can or cannot be integrated into other information systems. If the existing information systems is difficult to integrate, this means that it may no longer be viable to maintain the system due to its obsolete technology. Meanwhile, for existing information systems that is to be maintained, the appropriate interface needs to be established to facilitate integration with other information systems, or interface for the data warehouse. A new information system is being developed in stages according to the integration standard after which the previous information system can then be terminated.

3.5 Requirements Analysis of WIS

Requirements analysis is intended to collect information necessary for WIS development that shall help meet HEI business goals. Requirements are identified based on the internal and external needs of the HEIs.

3.5.1 Internal Requirements

Internal requirements are formulated according to the needs of the HEIs as well as their departments and relevant divisions. The requirement of each department and division may differ but in general will be adjusted by conforming to the existing procedure. Given the internal requirements, the following aspects should be taken into account during information system development:

- a. Web-based platform
- b. Supports changes including with regard to changes in the regulation and policy
- c. Allows automation of academic transactions
- d. Allows the customization of functions specific to the needs of departments and working units

3.5.2 External Requirements

External requirements are formulated based on government regulations applicable in Indonesia which in this case refers to the Ministry of Education, higher education policies and

strategies, reporting requirements and other general requirements. Given these external requirements, the WIS to be developed must:

- a. Be compatible with the National Higher Education Database (*Pangkalan Data Pendidikan Tinggi*, PDPT)³. Data and reports needed by PDPT as shown in table 3.1.
- b. Support standards and formats for academic program reporting purposes to the government (in this case to DGHE) such as the study program evaluation based on self-assessment (*Evaluasi Program Studi Berbasis Evaluasi Diri*, EPSBED⁴) report [72]. Data and reports needed by EPSBED as shown in table 3.2.
- c. Comply with QA standards for the accreditation of HEI issued by the National Accreditation Board for Higher Education.

Table 3.1 Data and reports needed for National Higher Education Database (PDPT)

Standard	Data and Reports Needed by PDPT
Content	Structure of curriculum, curriculum content, distribution of courses in each semester, courses sequence, and academic calendar
Process	Syllabus and lecture plan, sources of learning, classes activities, method of course assessment, quality of lecturing, result of monitoring and evaluation
Assessment	Students achievements of the targets set in core and elective curriculum as well as extra-curricular activities
Graduate Competency	Assessment of graduates' knowledge, soft, hard, and practical skill

Table 3.2 Data and reports needed for EPSBED report

Standard	Data and Reports Needed by EPSBED
Identity	The study program, university and foundation
	Students, Teachers and Graduates
Process	Teachers (courses taught on each semester)
	Students (courses enrolled and grades on each semester)
	Graduates (grade point average and length of enrolment)
	Curriculum
Statistics	Students, Teachers and Graduates

³ PDPT stands for *Pangkalan Data Pendidikan Tinggi*. This is Indonesian terminology meaning National Higher Education Database.

⁴ EPSBED stands for *Evaluasi Program Studi Berbasis Evaluasi Diri*. This is Indonesian terminology meaning Study Program Evaluation based on Self-assessment.

3.6 Architecture Used: Creating the foundation for WIS Development

Based on the previous analysis, this section shall explain on the IT architecture model which serves as a reference model for WIS planning, design and organizing within HEI. The proposed EA-MDA model exhibits all activities required to develop the architecture for HEI, beginning from vision construction to implementation and evaluation. Figure 3.1 presents the proposed model.

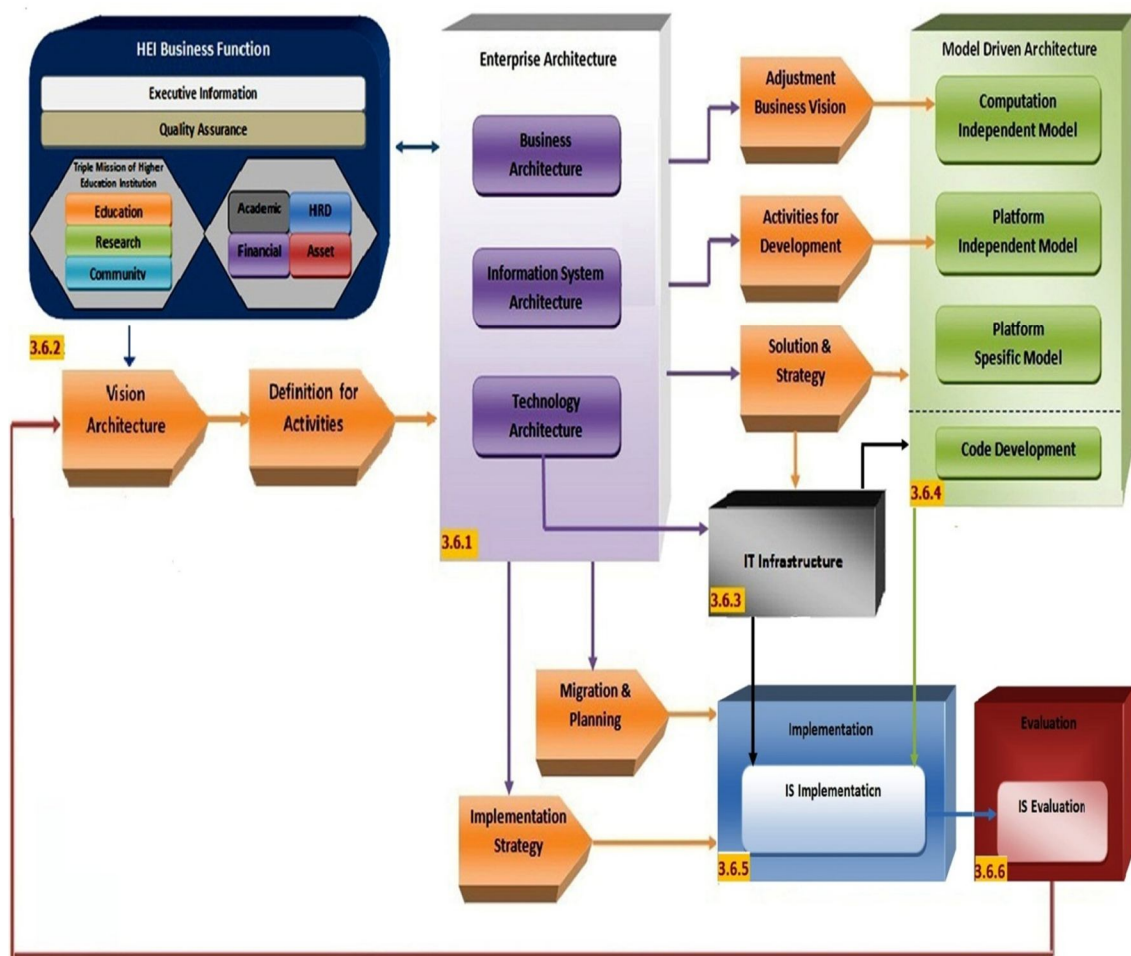


Figure 3.1 Information Technology Architecture Model for HEI

Vision and mission of HEI refers to the applicable government regulations. While the scope and requirements refer to the mission and vision of each HEI. Further business architecture, information system architecture and technology architecture refers to the requirements,

business processes and resources available in each HEI. Stages involved in this design process are critical to lay the foundation for the subsequent phase of designing the architecture or blue print, as output for the IT architecture model. By applying the IT architecture model, the HEIs can evaluate their existing information systems and technologies, while defining the need for a new information system and technology. Information system development for HEI shall then be based on the IT architecture model.

This IT architecture shall guide the institutions in developing an integrated information system to support HEI goals. This section will discuss all activities in IT architecture model in order to create the IT architecture and serve as an input for WIS development. We focus on the IT architecture while IS implementation and IS evaluation will be discussed in the next chapters.

3.6.1 Enterprise Architecture

3.6.1.1 Business Architecture

Business architecture in general describes the series of business activities, data and information found in the internal and external environments of the HEI. This phase is specifically meant to gain insight on the current situation of the business process before improvements are recommended by initiating the business architecture modeling process. Business architecture modeling may refer to business functions already analyzed previously specifically by considering the primary and supporting business functions of the HEIs. Business functions can be identified by using Porter's value chain approach. At this stage, modeling tools and basic methods such as Business Process Modelling Notation (BPMN) and UML can be employed to develop the desired model.

3.6.1.2 Information System Architecture

This section emphasis on how the information system architecture is built that includes data architecture and application architecture that will be used by the institutions.

a. Data Architecture

Data architecture refers to the identification of data that supports business functions as defined in the business model explained earlier. Management requires centralized and

integrated data sources in order to heighten coordination and synchronization of data management operations. Within the context of data architecture, university needs to avoid the repeated inputting of the same data. Data should be inputted only once and can subsequently be used simultaneously by relevant business processes. The data architecture currently developed by university still has weaknesses that need to be addressed before it can be integrated with data originating from other universities.

If data is to be integrated with other HEI, the data architecture create by an HEI will still have serious shortcomings. Similarly, setbacks are also evident in the reporting of important data such as information related to the teaching staff, students and other academic data that must be reported on a semesterly basis to the DGHE application. Several HEI data reporting systems are still manually operated, thus leading to various problems including the use of media unreadable to the application used by DGHE which requires the HEI to first make the necessary improvements. Furthermore, data cannot be changed immediately.

In this stage, data architecture is developed through the identification of the business function and organizational entity. Results of the identification process are then presented through an UML class diagram or a data functional matrix to illustrate the connection between business process and data entity through created, read, use and delete (CRUD) functions. The relation between each data entity with another entity is analyzed then compared with the list of data currently being managed by the system. This comparison is necessary to ascertain on whether data covered in the data architecture entity is based on data description.

b. Application Architecture

Application architecture focuses more on the planning of application needs and creating of application models. Application architecture required by the HEI is an integrated, online application that runs on a standard platform. In addition, management also requires a dynamic application and real-time system to ensure the presentation of timely and accurate information. Application architecture can be described through the application interaction matrix of business functions within the organization or the Technical Reference Model (TRM). Application is linked to business and organizational functions in order to keep track of the collective use of an application. The impact of application architecture towards existing

applications is analyzed. The appropriate solutions are required in order to determine on whether available applications are to be maintained or modified, integrated into other applications or new applications developed.

Based on a survey launched on 100 HEIs in Indonesia [2], there are at least 9 (nine) information systems that are largely established in these institutions : admission for new student, academic, e-learning, e-library, e-laboratory, e-curriculum, research and community service, alumni and carer, study program evaluation (EPSBED). All nine information systems are primary applications that should be available in each HEI to prevent disruptions in its business process. These information systems must be taken into account when formulating the application architecture required by the HEI consistent with the characteristics of HEIs in Indonesia.

3.6.1.3 Technology Architecture

Technology architecture seeks to identify technology platforms, analyze the use of current technology platform toward applications, and propose technology platforms related to university needs. The appropriate technology platform is selected by assessing current IT trends and developments such as trends in hardware, software, network, database, security and social network. Results of technology classification include the selection of viable technologies for technology platforms that shall support applications and the recommended technology development. The technology architecture model essentially describes how technology supports applications and user interaction when using the applications.

3.6.2 Support Activities

The support activities that will be required for developing IT architecture model in HEI is shown in Figure 3.1 as input and output of Enterprise Architecture, Model Driven Architecture and Implementation phases. Those activities will be influenced by the principles of the architecture used. Architectural principles is the early stage in realizing the architectural design activities to obtain a model or blueprint of information system in accordance with the needs of the HEI. The following is description of each activity :

3.6.2.1 HEI Business Function

Formulating the IT architecture model for HEI is inextricably linked to the analysis of the university internal and external business functions as regulated by the government and set out in regulations on higher education in Indonesia. Business function is divided into two groups: primary business function and supporting business function. The primary business function of universities in Indonesia generally comprise primary business activities grounded in the Triple Principles of Higher Education Institutions, namely education, research and community service. The supporting business function on the other hand contains supporting business activities that include academic administrative management, human resource management, finance management and asset management. All of these activities must meet the Quality Assurance and Executive Information System standards.

3.6.2.2 Vision Architecture

This stage involves the identification of management requirements, project scope and constraints, and the definition of advanced architecture and expected targets. These identified aspects are represented through the stated vision and mission, business goals, and business objectives. The output at this stage is the creation of the vision architecture to be used. To have an idea of the general universities vision in Indonesia, several samples of Indonesia's leading universities have been examined [73]. From this sampling it can be conferred that the stated vision of university in general is to establish themselves as leading seats of learning of national and international reputation by building on their respective core competencies. Their mission statement on the other hand aspires to deliver first-rate educational processes based on the Triple Principles of Higher Education Institutions.

3.6.2.3 Definition for Activities

Prior to starting work on the subsequent architecture model, it is necessary to document and define all data needs, data organizing applications and data sharing requirements in running the business process, and existing technology platforms employed within the enterprise. Documentation shall become the basis for architecture modeling and the following implementation plan. Based on the primary and supporting activities of the HEIs, this stage

shall define activities necessary for documentation purposes that cover literatures, surveys and interviews with relevant parties.

3.6.2.4 Adjustment Business Vision

Architectural models obtained in the previous stages will be reviewed at the next activity. The modeled business and application architecture in the previous stages is then re-assessed with regard to its compatibility with university business vision. This stage maps out the relation between application architecture and the achievement of the vision formulated earlier. The output of this activity will be the basis for the development of the system, especially at the CIM level in the MDA approach.

3.6.2.5 Activities for Development

WIS development is also conducted based on the guidelines from the list of WIS development activities. This list is necessary to provide a more focused WIS development process, beginning from preparations for needs assessment to the eventual development of a systems design for use by the higher education institutions. The output of this activity will be the basis for the development of the system, especially at the PIM level in the MDA approach.

3.6.2.6 Solution and Strategy Development

This activity is intended to identify main issues and seek viable solutions. From the results of the analysis on the current situation of the higher education institution and the architecture model produced, a gap analysis is conducted. Results of this gap analysis can become the solution and strategy for resolving issues. This stage is also performed to determine the priority of application development. These solutions and strategies in turn can serve as input for the subsequent process of developing a more comprehensive WIS and technology infrastructure need.

3.6.2.7 Migration and Planning

The migration strategy for a new WIS is required prior to the implementation. In this stage, a gap analysis is essential on the resource base, including changes that may arise upon the implementation of the system. In addition, it is also equally important to conduct an impact

analysis of the new application architecture and the decision-making process towards new IT investments in order to ensure a complete planning process with regard to systems migration and implementation.

3.6.2.8 Implementation Strategy

This stage involves estimations on human resource, implementation schedule, costs and benefits of the plan and time required. This includes the formulation of recommendations for each development implementation, determining the implementing organization and guaranteeing the compatibility of systems development with the desired architecture. Risk management is another essential component that needs to be incorporated into the implementation plan in order to minimize potential risks, such as risks related to human resource, unmet implementation schedule and others.

3.6.3 Technology Infrastructure

The gap that might occur between the current (as-is) architecture and propose (to-be) architecture can be eliminated by means of fulfilling the needs of technology and information system. This stage is conducted through procurement for technology support needed to encounter the needs of the technology of addition or optimization technology.

3.6.4 MDA based WIS Development

The web information system development process is based on the MDA and SOA concept. Metamodeling defines the structure of models, and in an abstract way specifies the construct of a modeling language and their relations. Through this approach, it is necessary to determine a metamodel that suitably represents the web application. During the design phase, modeling is used to define requirements and provide model details at various levels. Next, model transformation is carried out in accordance with transformation rules. The models provide support for testing prior to implementation and also contribute to automatic code generation.

3.6.4.1 Computation Independent Model Level

This stage is associated with an enterprise architecture particularly business architecture that already proposed in the previous stage. Business models need to describe the environment in

which the system will be used, with no direct orientation on how it will be implemented. It also specify the requirements, use cases and the system's main flow from the customer's perspective.

3.6.4.2 Platform Independent Model Level

In the CIM level and PIM level will be use business model, data and application model from previous results on the phase of business architecture and information system architecture however in a more detailed form. Modeling the systems process and structure can be done through these models in an independent way of the technological details for their implementation. The processes are captured by UML activity diagram and structures are capture by UML class diagram. These models can be modified or transformed into PSM that express the specific implementation process.

3.6.4.3 Platform Specific Model Level

The models are then transformed into a PSM based on web as target platforms. Templates need to be defined in order to automatically generate the web application and conduct model transformations. Based on the models and templates identified earlier, the code generation process can proceed to automatically produce the source code. In this case, web application is implemented through the PHP scripting language and additional technologies, such as the web server and database server. All files generated are placed into the framework. The remaining part are created manually to finalize the entire process.

3.6.5 IS Implementation

The implementation stage is where the WIS already developed in the previous phase is implemented. Through systems implementation, the outcomes can be operated and utilized optimally according to needs. At this stage, it can be determined on whether the system that has been developed can operate according to plans, and whether it can be utilized by the end user and meet the expected objectives. An evaluation is also conducted during this stage on shortcomings during the system development process, including a deficient requirements analysis, not modeled and unpredictable factors. As such, the necessary improvements can then be made to rectify these inadequacies.

3.6.6 IS Evaluation

Evaluation is an important aspect necessary for gleaning information on the extent to which the objectives of an WIS has been successfully achieved. Through evaluation, feedback will also be generated, crucial for improving WIS quality in the future. The main constraint in conducting evaluation is in determining the evaluation criteria, evaluation parameter and methodology for establishing the evaluation framework. Recent developments in IT, particularly with regard to the internet, have made it even more difficult to gauge the level of success and effectiveness of an WIS compared to earlier conditions. WIS has brought forth a different set of conditions and factors that need to be considered, such as direct user interaction unrestricted by time, distance and place.

3.7 IT Architecture : A Case Study

This section shall discuss on the application of the IT architecture model recommended above to support WIS development at University of Lampung (Unila) as the case study. Unila is a state-owned university in Indonesia that runs 8 faculties, 54 undergraduate study programs (diploma and bachelor's degree), 12 post-graduate study programs (master's and PhD degree) and currently serves 28.116 students, 1120 lecturers and 707 administrative staff.

Data and analysis on university activities related to its Business Function, Vision Architecture, and Definition for Activities required as input for modeling Unila's enterprise architecture are entirely based on current data and conditions necessary for Unila.

3.7.1 Business Architecture

For the modeling of the business process, Unila's business functions are identified and its organizational structure documented. These identified business functions are then linked to the working units in order to determine the responsibilities of working units in relation to a business function. Business functions are identified by using the value chain model that classifies business areas into primary activities and supporting activities of the enterprise. From this model, it can be concluded that Unila has six supporting business processes and three primary business processes. Three primary business processes related to education,

research and community service while six supporting business processes related to academic administration, human resource development, finance, asset, public relation and IT/IS management. The business process of Unila can be seen in Figure 3.2.

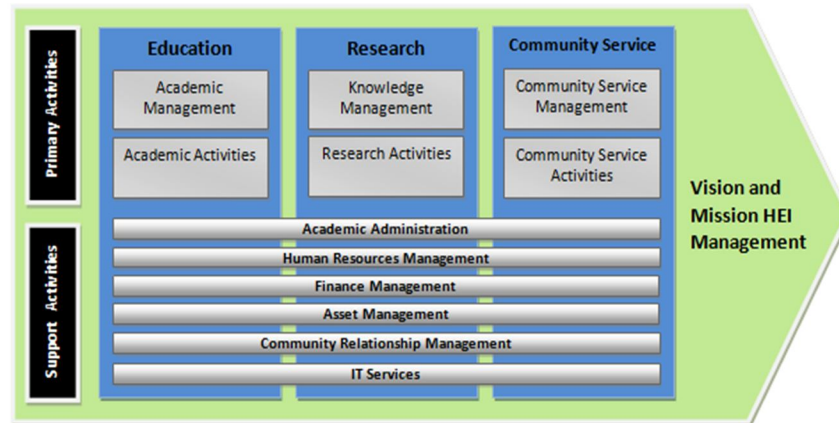


Figure 3.2 Unila's business process

3.7.2 Information System Architecture

3.7.2.1 Data Architecture

Data architecture defines the main categories of data needed to support business functions. It creates from the business architecture explained earlier which refers to data required for facilitating three primary activities and six supporting activities. This phase is prepared by identifying all the components of the data that will be used by applications based on defined business functions. Identification is through by determining candidate data entities, defining data entities, and creating relationships between business functions and data entities. The analysis results of planned data architecture of Unila can be seen in table 3.3 and Figure 3.3.

Table 3.3 Analysis results of planned data (to-be)

Business Function		Data
Primary Activities	Education	New student, Study program, Examination Categories, Registration, Student, Class, Curriculum, Course, Laboratory, Grade, Teacher, Advisory, Practical/Final work, Tuition, Graduation, EPSBED, PDPT, QA.
	Research	Researcher, Research, Research Categories, Publication, Publication Categories, Activity, Activity Categories.

	Community Services	Leader, Community Services, Community Services Categories, Publication, Publication Categories, Activity, Activity Categories.
Supporting activities	Academic Administration	Academic Administration, Academic Activity, Academic References, Teaching Learning.
	Human Resource Development	Teacher, Staff, Education and Training, Qualification, Preferment, Status, Document.
	Finance	Financial Budget, plan, accounting and audit.
	Asset	Inventory, Acquisition of assets, Maintenance of assets, Extermination of assets.
	Public Relation	Academic community cooperation, Institution cooperation, Industry cooperation.
	IT/IS Management	IT/IS assets, IT/IS operational, Services, Security.

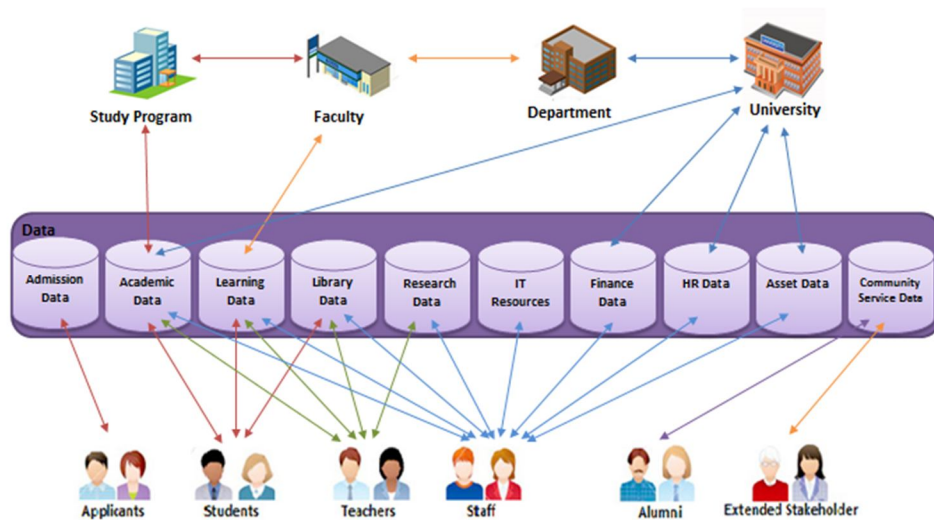


Figure 3.3 Data Architecture

3.7.2.2 Application Architecture

Application architecture refers to current (as-is) applications and planned (to-be) applications. An analysis is therefore necessary on ongoing applications to look at the business process and needs of working units. Analysis results on existing applications based on faculties, departments and working units responsible for managing the applications. In the development of current applications, Unila have three developments options, e.g., open-source based development, own-source based development, and for some special purpose the application

already be developed by the government and ready to be implemented. Available application technologies mostly use PHP and database and Oracle. A few other technologies still operate through Novel Netware by using DBF database. Websites use PHP and MySQL.

Unila has an as-is application based on the working unit responsible for the information system. The status of Unila's existing information system includes: (a) information system applied by the university, department and relevant division, but are not integrated with each other, (b) non-web-based information system that needs to be entirely rewritten in order to introduce a new functionality, (c) web-based information system, but without the web services required to integrate with other systems. A current major concern is to establish an integrated system where all existing information systems are interconnected.

Analysis results on existing applications inform decisions on strategies to be implemented for WIS integration. Subsequently, to identify future application needs, a Strengths, Weaknesses, Opportunities, and Threats (SWOT) and Critical Success Factors (CSF) analyses are conducted to determine the future WIS solution model appropriate for the business process carried out by Unila. Figure 3.4 shows the analysis results of planned application (to-be) required by Unila. There are classified in the Academic management, Knowledge management, Resource management, IT Services, and Community Relationship Management (CRM) to support Quality Assurance and Executive information that can be seen in table 3.4. Applications consists of existing applications, new applications and Integrated applications

Table 3.4 Analysis results of planned application (to-be)

Business Function	Domain	Description	Applications	
Primary Activities	Academic Management	Applications related to academic process	Admission new student, Graduation, Practical/Final work, Laboratory, Teacher evaluation, Academic, EPSBED, PDPT	Integrated Academic
	Knowledge Management	Applications related to research and community service	Research, Community service, Digital library, E-journal, E-learning, Knowledge	Integrated Knowledge
Support Activities	Resource Management	Applications related to human resources,	Human resource, Finance, Asset, Procurement, RKAKL,	Integrated Resources

		finance and asset	Repository, State Property, Cooperation, SERDOS	
	Community Relationship Management	Applications related to public relation/ community relation	Portal website, Help desk, Career, Alumni, Parent's student, Forum, Industry	Integrated Public Relation
	IT services	Applications related to IT services	WIFI/Proxy, Email, Firewall, SMS GW, DNS, VOIP, BW manager, Network monitoring, Video conference, File sharing	Integrated IT Services

Figure 3.4 presents a proposed application architecture consisting of applications required for primary and supporting activities in Unila. It consists of Executive Information System (EIS), Quality Assurance System, applications in five domain managements, Single sign-on (SSO) and Data warehouse.

The Executive Information System (EIS) refers to a system that provides information for executives on HEI overall performance. EIS offers a more flexible representation of the university that can be easily understood by executives in looking at the root cause of problems and the most viable solutions, and allows speedy access to the necessary management information and reports.

The Quality Assurance System is useful for tracking QA-related information released internally and externally by the HEI. In addition, the system facilitates an online internal evaluation process of study program through the Semester and Annual Internal Evaluation mechanism simultaneously completed by all programs according to the predetermined time frame.

The recommendation to develop an integrated system is deemed necessary given the need for each application to be interconnected with each other according to the business process. In Academic Management for example, the new student intake system shall be connected to the academic information system, mainly with regard to data consolidation of students who have passed the admissions selection process. Not only does it deal with the new student intake system, the academic information system is also connected to the laboratory and e-learning system under Knowledge Management. It is therefore recommended that an integrated academic information system be established capable of providing and delivering academic and student-related data and information.

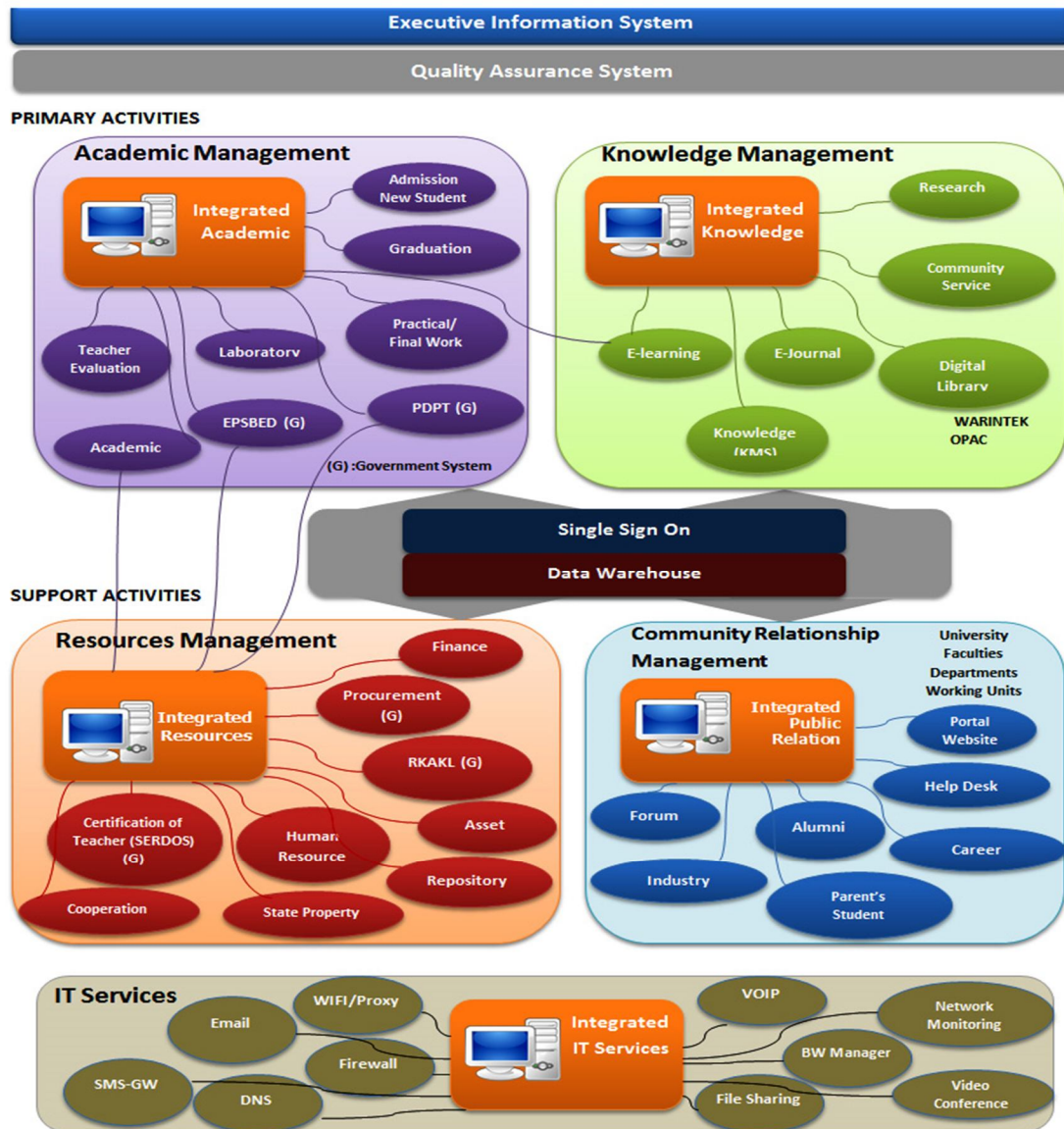


Figure 3.4 Application Architecture

Single sign-on (SSO) refers to an access control mechanism that allows users to operate the system without the need for repeat user authentication. Authentication is a mechanism that identifies and authorizes a user either as a guest or registered member of a system. Before SSO can be applied, the authentication needs should first be clearly defined. This is because not all users and applications will need this system. The need for SSO is mainly for ensuring users the 'ease' to access an application without having to spend time re-entering their authorizations. It

also helps reduce complexity because users need not memorize different access codes for every application.

Data warehouse is required for the storing of historical data separated from the operational application. This is essential to ensure better application performance that allows for the generation of various types of information, such as for reporting purposes and data analysis integrated into available raw data. Data warehouse makes it possible for the integration of different types of data from various applications or systems. This will guarantee the availability of a single-door access mechanism for HEI management to obtain information and analyze it for making decisions on the institution's development and expansion. An integrated data system will be useful for the delivery of relevant information that can be viewed from different angles from which its level of detail can be adjusted accordingly. With data warehouse support, decisions and strategies can be made and formulated by executives because data warehouse handles external data from recognized sources from which data can be examined either in detail or as a whole. The data warehouse system allows for faster, and more effective and efficient analysis of data.

3.7.3 Technology Architecture

This particular phase is intended to identify the current technology platforms and how they are applied with regard to applications, and determine the recommended technology platforms that Unila requires. Unila's existing technology platforms identified include its data center equipped with 32 servers, 2000 desktop computers connected to the Local Area Network (LAN), and data storage in the form of SAN (Storage Area Network) [74].

Its intranet infrastructure that links inter-nodal working units relies on a switching system applicable up to the faculty level whereby all switching functions are done through the manageable switch mode. Fiber optic as the main media backbone has 1 Gbps in channel bandwidth. Wireless media is also used as the network infrastructure, although several study programs (30% from all study programs in Unila) are still using cables and have yet to employ wireless technology.

The existing technology platforms are largely support of the proposed application but still need optimization to improve the performance. From Unila's needs assessment, the technology required by the institution is a network technology that connects applications and allows easy access for end users. Technology platforms recommended for development must be those that adopt recent technological trends such as the latest in hardware, in memory computing (IMC), network, cloud computing, web 2.0., e-Learning 2.0, mobile computing, Internet of Things (IOT) and social network.

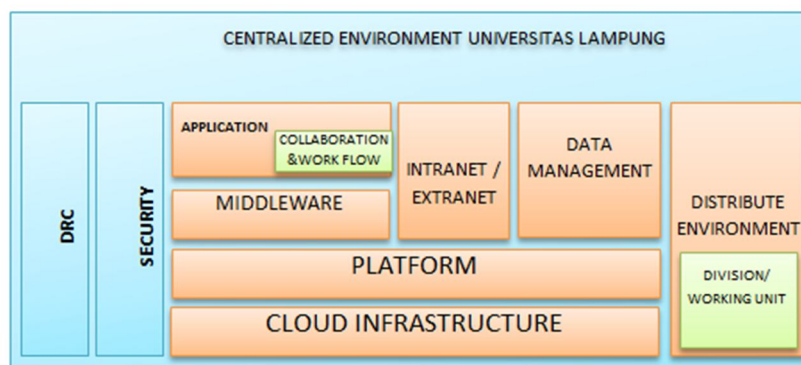


Figure 3.5 Proposed technology architecture

Figure 3.5 presents the proposed technology architecture. This architecture focuses on software and hardware design encompassing cloud infrastructure, platforms as required by the HEI, middleware, networks, data management, non-application software, physical facilities to support applications and business processes required by the HEI, as well as the Security and Disaster Recovery Plan (DRP). In addition to information system architecture, the IT architecture is essential for supporting WIS implementation. However, this study shall focus on information system architecture. Subsequent activities for WIS development shall be based on the planned information system architecture.

3.7.4 Solution Strategy for Bridging the Gap

Subsequent activities as follow up to architecture modeling include Adjustment Business Vision, Activities for Development, and Solution and Strategy Development (as can be seen in Figure 3.1). Based on the results of an analysis conducted in the previous phase, Unila needs to

integrate existing applications, integrate data from various working units, and carry out a combined integration with the new application to be developed. Application integration is the solution for Unila's problem given the need to continue with the maximum possible use of the database and web services in existing applications to the extent that each working unit can share data and processes without the need to make changes to the application or data structure.

Based on the current situation of existing information system in Unila and HEI in general, solutions and strategies are formulated for integration purposes with regard to the presentation (user interface), applications, services and data. The integration is also performed with the data on DGHE's applications. Data communication between the university and DGHE through existing university ICT network- INHERENT. There is a data integrator that serves to consolidate and integrate data in university with the data on DGHE. Figure 3.6 presents the purposed integration model architecture. The solution model produced can connect existing applications with other ongoing applications as well as with new applications without affecting the performance of the respective application.

As can be seen in Figure 3.6., the users of a web information system in Unila can be divided into five groups : the group of applicants, then the group of students and teachers who involved in teaching learning, the group of staff (administration and management), the group of management who involved as decision maker, and the group of alumni and extended stakeholder. For all user groups the relevant master data should be kept that are needed in the particular business processes. Each group can access related applications, services and data from internal institution or external institutions such as from the government's resource.

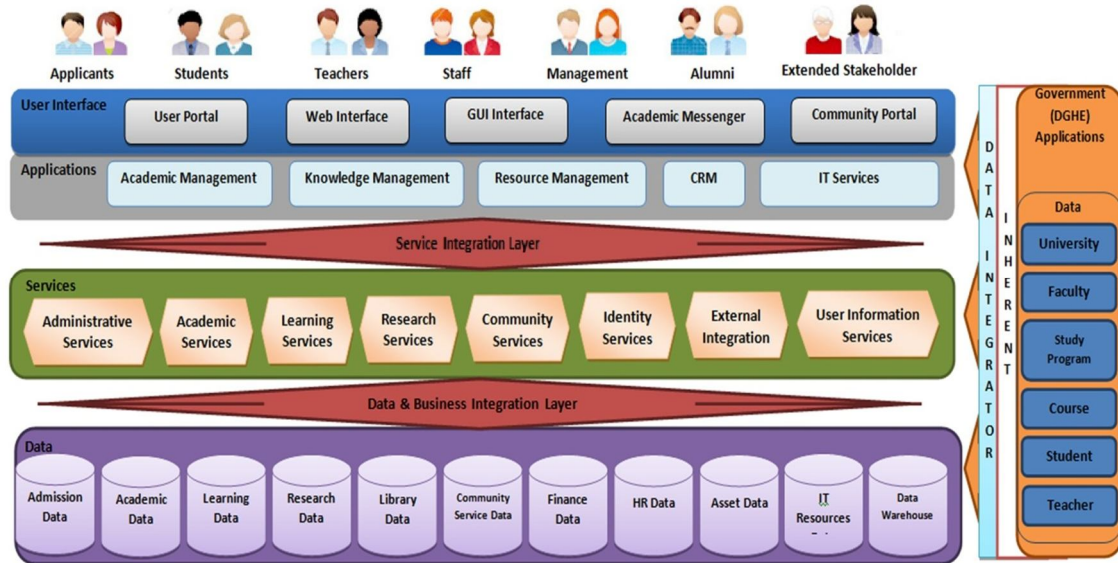


Figure 3.6 Integration Model Architecture

3.7.5 Migration and Planning

The purpose of Migration and Planning phase is to plan the migration process or the transition from the old system to the new system to facilitate the information systems to be developed more focused. One of the most essential aspects of the migration process is to minimize the risk. The implementation of Information Systems is expected minimum possible risk. It can be prepared in the following approach :

- Information systems should be well documented and structured that it can easily be tracked.
- Information systems should be implemented in parallel approach with several existing applications.
- Provide training for users.
- Provide dissemination to all stakeholders.

3.7.6 Implementation Strategy

The implementation strategy is intended to prepare a plan for applying the architecture already established- in this case referring to the data architecture and application architecture-

so that successfully implemented. Besides that plan, factors that determine implementation success are as follows:

- (a) HEI management involvement, support and commitment that should consistently remain focused on information system development
- (b) The need to establish a working unit that specifically takes responsibility for the implementation
- (c) The information system development process should be made known to all working units in order to ensure constructive contribution for subsequent information system development.

The relationship between applications and data entity is created for data sharing in the application architecture. The scheme can also be used for determining the chronological order of applications to be developed in which applications that generate data should take priority for implementation compared to applications that will use data. Other factors that need to be considered in determining the order of implementation include need, benefit, risk and impact towards the organization.

Concerning WIS development, based on Unila's needs assessment, the domains that support primary activities (Academic management and Knowledge management domain) shall be given the highest development priority. The next chapter focuses on the development of a web information system to integrate the existing systems in educational institutions. The analysis results in Academic management domain mentioned in section 3.7.2.2 will be used as a case study for WIS development in Unila.

3.8 Chapter Summary

Educational institutions require a proper standard and model of IT architecture that can be implemented to enhance alignment between business strategy and existing information technologies. Developing the required model is a complex task, it needed a comprehensive methodology and easy to apply. A combination of the EA and MDA concepts can be one of the solutions to overcome the complexity of building a specific IT architecture model for HEIs. The purpose of this chapter is to formulate an IT architecture model that provides clearly defined directions with regard to inputs and outputs for EA development activities. The proposed

model uses the HEI business function that set out in government regulation and seven support activities as input and output of EA, MDA, IS Implementation phases. By applying this model, every HEI can evaluate their existing information systems and technologies, while defining the need for developing a new information system and integrating the existing ones.

IT architecture model based on EA-MDA has been developed according to the condition and needs of national universities. This model is then applied to University of Lampung as a case study. The outcome is an IT architecture that can be implemented to support integrated WIS development in Unila. The IT architecture consists of the Business Architecture, the Information System Architecture and the Technology Architecture. For Business Architecture, the analysis results of business function show that Unila has three primary business processes and six supporting business processes. For Information System Architecture, the analysis results of planned data are all the data that will be used by applications based on defined business functions. Meanwhile the planned application required by Unila are classified in the Academic management, Knowledge management, Resource management, IT Services, and Community Relationship Management (CRM) to support Quality Assurance and Executive information. From a needs assessment conducted on data, application and technology, some 40 applications have been identified that should be available in Unila. Most data and functionalities are in fact already available in existing applications but current data and applications are redundant, inconsistent and isolated from certain working units. Enterprise integration can shorten the life cycle of systems development as existing applications and information can be reused and there is no need to repeatedly create the same business processes, applications and data.

Chapter 4 MDA and Integration of Web Information Systems

The demand for developing new WIS and changing the existing WISs has to be done within an acceptable time by business conditions. Likewise the development of integrated information system is also considered by educational institutions due to the number of existing information systems with different platforms and database. On the other hand, the development of WIS are still not able to pursue those needs. Generating web application automatically arise with a promise to solve these problems, namely the model-driven based WIS development that is relatively fast and ease of system maintenance. In this chapter we propose a development method of a web information system through the adoption of the Model-Driven Architecture (MDA) and Service-Oriented Architecture (SOA) approaches that can be used for generating web application automatically.

4.1 Chapter Introduction

Business enterprises and educational institutions need to keep abreast with new technologies that evolving rapidly, especially in the field of web information development, in order to adopt the necessary technological change which may require the integration of existing systems. Technology and information system in an educational institution vary widely, frequently not able to share resources and ensure synchronized operations among existing systems. Web information systems are therefore crucial to support the academic process. In view of this, interoperability becomes a very essential issue as it is the basis for ensuring the integration of different information systems. The integration of existing systems makes it easier for users to

access and manage available data. Furthermore, information can be immediately retrieved which in turn will lead to more accurate and timely decision-making [75]. This optimal use of available information through an integrated system will certainly increase the effectiveness and efficiency of academic services provided by educational institutions, thus ensuring the highest level of service for students, teachers and other constituents.

Considering this background, a web application that serve as an interface (called web application interface) is required to enable the various information systems through the use of web service technology. Given the increasing number of available web services, web application development for integration purposes has become increasingly in demand. A web application that is intended in this paper is a web information system that facilitates web-based access to complex data and interactive services for users. One of the advantages of a web application interface is that it helps integrate existing university-wide information systems and has the far-reaching impact of improving the education field. In addition to being flexible, the interface must prevent dependency, and should allow for relatively easy modifications and revisions in the future. To support management's work, such application provides the necessary mechanisms for effective decision-making, strategic planning and control. In order to become a reliable system, the following requirements are essential: improved communication, self-service functionality, content management and publishing, reporting and statistics, information extraction, as well as data security and privacy.

To ensure the broadest usage applications and the ability to effectively respond to ever changing business needs, it is advisable for an institution to develop a new web information system by reusing existing resources, web services and data. The development of a new web application interface is however the preferred option rather than the modification of existing applications. The latter approach is discouraged because developers may not be comprehend with existing applications development. Even if they are open-sourced applications which allow for modifications, taking the modification technique may generate a different set of problems. For example, problems may arise when operating the upgraded version. Using an upgraded version through adjustments to the application may result in incompatibility with previous versions. Furthermore, solutions to problems that the modified application was intended to address may not be applicable for problems of more general problem. In contrast,

developing new web application can prevent the emergence of problems stemming from changes made to existing applications, while offering new functionalities.

This work describes a development of new web information system that serve as an interface between existing information systems through the adoption of the MDA and SOA approaches.

4.2 Web Information System Development

4.2.1 Web Design

An integrated approach will combine sub-systems into a single web application system and guarantee that all functions of the sub-systems work as a cohesive system. With regard to the major functions of existing systems, systems integration should continually build on the consistency of available systems in order to ensure that they continue to operate accordingly. The challenge is to design a mechanism that integrates these existing systems with the least effort. The development of a new web system – in this case the web application interface – allows for the integration of systems based on functionality by connecting existing sub-systems while remaining anchored to the previous sub-system architecture. This method has the advantage of furthering the integration process because it only involves several relevant development entities. Furthermore, it allows for the implementation of a new business process or functionality that could not be done before due to constraints in the existing systems.

The development of web application interfaces largely aims to provide integrated data through web services from existing applications that emphasize more on different functional aspects for various purposes. Such web application interfaces make it potential for two different applications to interact simultaneously, ensuring that all applications work in a synchronized method. Users can easily interoperate across applications without the need to know how contents are stored and how services are implemented. Figure 4.1 presents an illustration of user access to the existing web information systems (WIS A and WIS B) and web application interface.

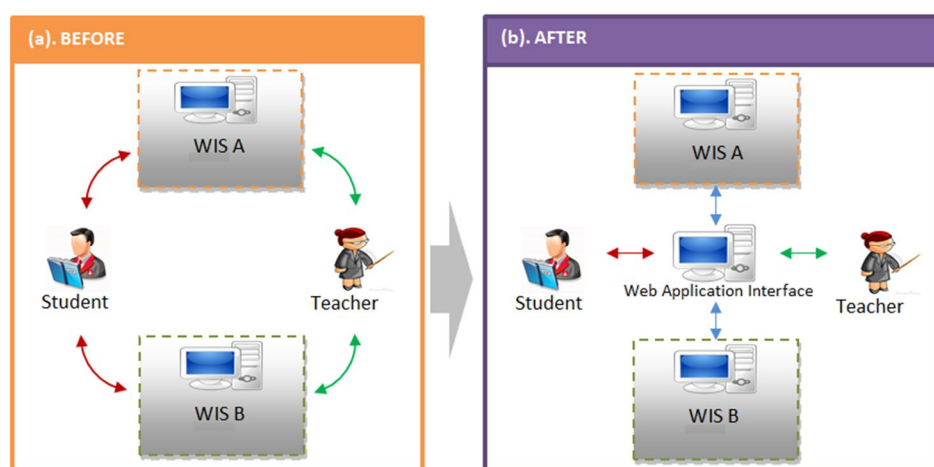


Figure 4.1 Illustration of web application interface

As can be seen in Figure 4.1 (a), the WIS A and the WIS B have major functions that should be accessible to student and teacher. For some specific functions, the WISs have some common data however they cannot interact each other. It is troublesome as users must create the similar transaction twice from different systems. Similarly, other problems resulting from asynchronous data that may occur.

In this case, additional web application performing as the interface to enable WIS interacting each other is required, as can be seen in Figure 4.1 (b). It is also to solve the problem of data consistency and ease of handling new business processes between two existing information systems. Through the integration of different systems, data previously entered in one system is also automatically available in other systems. The actions initiated in one system will also affect others therefore the users should not make a transaction twice while using them. However, the existence of a web application interface is not intended to disregard the user access both existing WISs. The WIS A and the WIS B are unavoidable to access by the users due to their major functions in academic activities.

4.2.2 Data and Web Services as Integration Path

The development of a web application interface should initially begin with the identification of business processes and system capabilities of each existing system. Redundant processes and information should be discharged to avoid the duplication of input data, divergence in

information flows or inconsistent processes. This step is a more detailed form of the results that have been obtained on the Business Architecture phase as described in the previous chapter.

The following step is data and information mapping from available documentation to determine data variety and potential data combinations. This generally involves the collection of detailed information, i.e., name, format, attribute, sources, level of security, life time and users of data. Data-related web applications that will be integrated refer to the analysis results that have been organized on the phase Information System Architecture as described in the previous chapter.

Web services can then be defined or selected from the list of web services on existing systems. The expected output is the mapping of data and service catalogue containing the list of web services needed from the existing systems. The application design for this particular case which follows the MVC pattern as shown in Figure 4.2. Through the integration of different systems, data previously entered in one system is also automatically available in other systems.

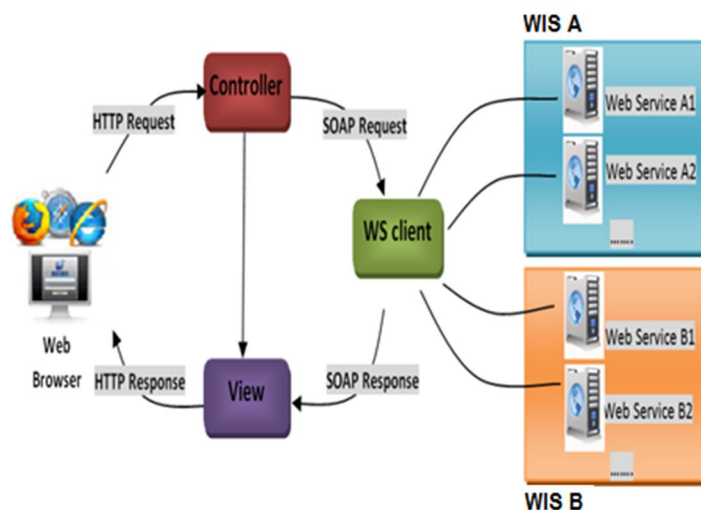


Figure 4.2 Web application interface design

The controller will handle the user's request. Then it calls to web service clients, selects and covers the response views. Web services are provided by existing applications.

4.3 WIS Development based on MDA-SOA

MDA SOA is the combination of a service-oriented architecture concept with a model-driven approach. This is a highly powerful method that can be applied in WIS development for an enterprise such as higher education institution. Nevertheless, the direct implementation of both approaches will not be an easy task. In this study, we have adopted and customized these approaches to suit the university's specific needs. We propose our method for the development and the integration of a web information system for educational institutions.

4.3.1 The Proposed Method for WIS Development

The proposed method in this work will identify a set of models at different abstraction levels. This can start from a high-level business model to a lower-level form. The model is associated with enterprise architecture model particularly with business model, data and application model from previous chapter results however in a more detailed form. The models is created first based on the use case, activity diagram and class diagram. The model is then transformed into a PSM based on web as target platforms. The PSM is then transformed into a code based on three types of code: PHP (MVC pattern), WSDL and DDL SQL (Standard Query Language) scripts. By using the method, web application systems can more easily and systematically be analyzed, designed, and generated and, thereby, increase web system development productivity. The WIS development using our proposed method comprises of five main steps as shown in Figure 4.3 :

4.3.1.1 Creating business models

Business models describe the environment in which the system will be used, with no direct orientation on how it will be implemented. It also specify the requirements, use cases and the system's main flow from the customer's perspective.

4.3.1.2 Creating process models

Modeling the systems process and structure can be done through these models in an independent way of the technological details for their implementation. The processes are captured by UML activity diagram and structures are capture by UML class diagram. These

models can be modified or transformed into PSM that express the specific implementation process. For this particular study, the MVC pattern is applied as a target technology for the generated web application.

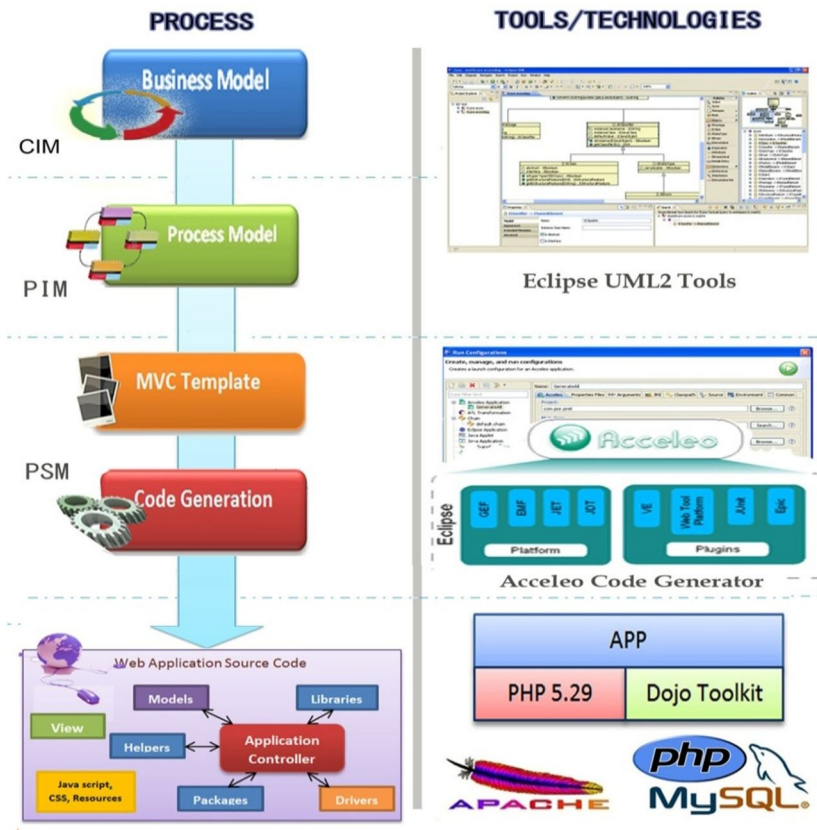


Figure 4.3 The WIS development method

4.3.1.3 Defining templates

Templates need to be defined in order to automatically generate the web application and conduct model transformations. They state transformation rules for changing a given model-to-model or model-to-text transformation. These templates are designed to generate the model to DDL scripts, WSDL and the web application based MVC. MVC formats are to be presented in a web application as PHP files output. It comprises on MVC_Controller that functions to receive requests, invoke the MVC_Model to perform the requested operations

and send data to the MVC_View. The defined templates also facilitate the generation of highly specific PSM from PIM.

4.3.1.4 Code generator

This step is regarding the generation of code for target web application using the Model-to-Text transformation. Based on the models and templates identified earlier, the code generation process can proceed to automatically produce the source code. In this case, Aceleo will be applied for creating the templates and producing a PHP-based web application [76].

4.3.1.5 Web application source code

Web application is implemented through the PHP scripting language and additional technologies, such as the Apache web server and MySQL database server, as the target environment for deployment. All files generated are placed into the framework. To prove our approach of generating web applications based on a MDA, the CodeIgniter [77] PHP framework is used because it is currently a popular Web oriented MVC implementation. Most Web applications are based on the MVC pattern for separating data and their presentation. All files generated are placed into the framework. The remaining part are created manually to finalize the entire process

4.3.2 WIS Development Process

This section presents the WIS development process. The structure of web application interface is explained, while designed templates and their role in the process of code generation are elaborated in more detail. By using the defined templates and models, the code generator creates all application source code based on MVC pattern, DDL and WSDL. Handwritten (manual) codes need to be added after generation for complex business logic and graphical user interface (GUI) parts. Then the result of generated and handwritten code are tailored to the framework for system implementation. The development of web application interface is combined with a framework CodeIgniter, leading to faster PHP web application development. Frameworks are designed to simplify programming and encourage code reuse by setting forth

folder structure, documentation, guidelines and libraries. To finalize the process development the web application has to be configured. The detail of development steps as shown in Figure 4.4.

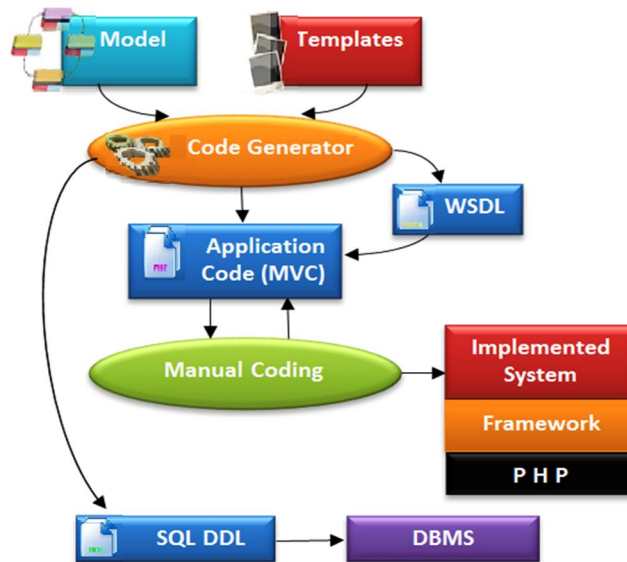


Figure 4.4 Detail of WIS development process

4.3.2.1 The Defined Metamodel

Metamodeling is one of the main issues in MDA. It defines the structure of models, and in an abstract way specifies the construct of a modeling language and their relations. Through this approach, it is necessary to determine a metamodel that suitably represents the web application. The metamodel is applied to generate platforms that adopt the MVC pattern. It provides the major specifications guiding the modeling of the web application based on a high-level design and facilitates model association related to the target code generation.

A specific metamodel acquaintance of UML metamodel is created for this modeling process. Based on the result obtained from the modeling process, this process expects to generate artifacts that may help create the necessary web pages. Figure 4.5 shows a simplified form of the metamodel. As can be seen in the following Figure, the metamodel is based on three packages: the MVC_Model, MVC_View and MVC_controller package. The MVC_Model package represents the web application domain, while the MVC_View package covers views

and concepts related to defined meta-classes, and the MVC_controller package refers to the control part of the MVC pattern.

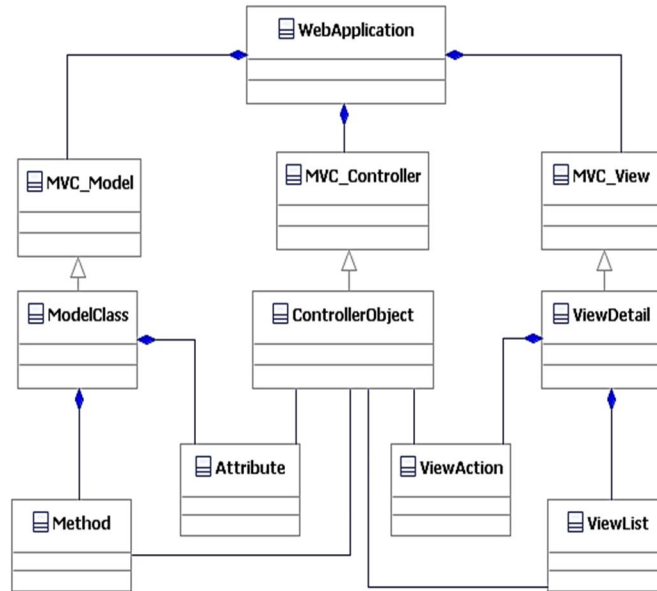


Figure 4.5 A simplified form of the metamodel

4.3.2.2 Defined Templates

The modeling process only constitutes half of the entire MDA system. Templates to produce the target system are equally important. The usefulness of the templates is to transform a given model to text. The model along with the templates will become an input for generating web application. We defined transformation templates include for generating DDL script to create the tables, for generating WSDL and for generating web application based MVC. The MVC templates, consists of templates for the MVC_Model, MVC_View and MVC_Controller implementation. A web application interface developer has the option to appropriately select from available templates according to the design and data that the developer wish to display. The list of templates created for is shown in Figure 4.6 and also described in Table 4.1, Table 4.2 and Table 4.3 respectively.

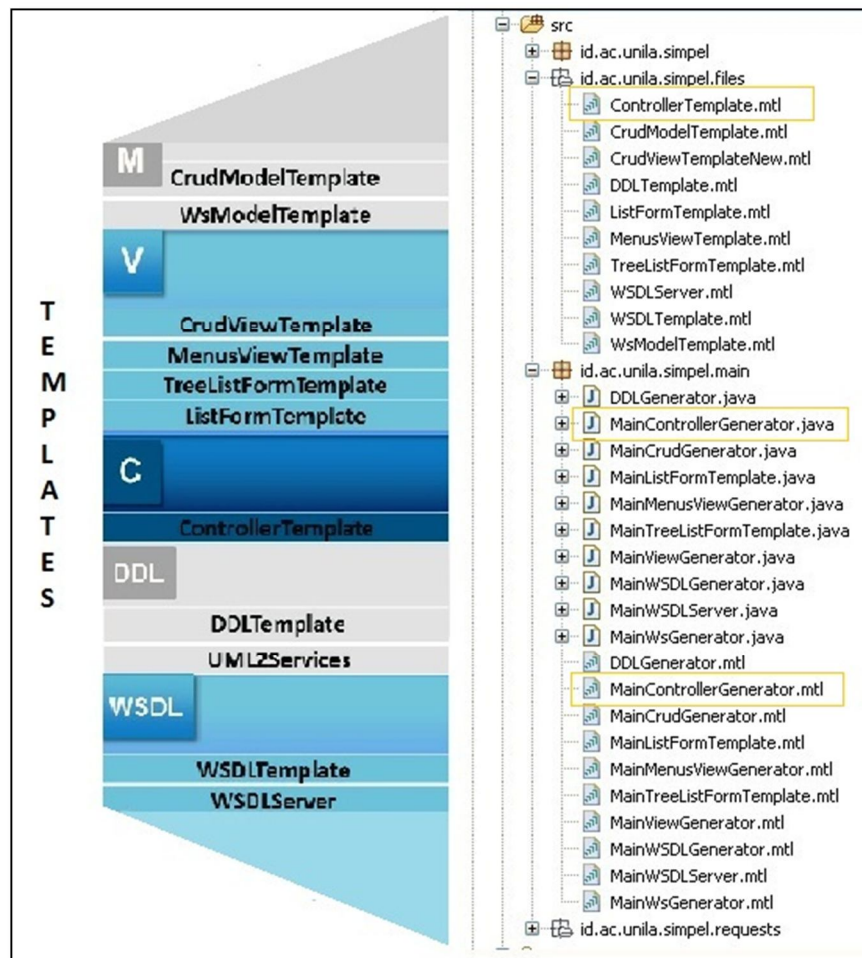


Figure 4.6 Templates for generating files

As can be seen in Figure 4.6, all files .mtl inside folder.files (id.ac.unila.simpel.files) are templates that have already defined to WIS development. In this case, 7 defined templates for generating web application based MVC, 2 defined templates for generating DDL dan WSDL respectively. When Aceleo will generate the code, it requires a MainTemplate (Main Module). Main template consists of one or more file templates (from folder.files). In the Main template, all defined templates that will be used are selected to call in the code generation process. For each template (xxx.mtl file) will have Main Template (Mainxxx.mtl file and Mainxxx.java file). Mainxxx.java file functions for running Mainxxx.mtl file. For example, to generate the code that deals with MVC_Controller, ControllerTemplate.mtl file must be run through the MainControllerGenerator.mtl which is called by MainControllerGenerator.java.

Table 4.1. MVC Templates

Template	Name	Description
MVC_ Model	<i>CrudModel Template</i>	A template for generating code which related to database operation (Create, Read, Update and Delete)
	<i>WsModelTemplate</i>	A template for generating code which related to web service functions
MVC_ View	<i>CrudViewTemplate</i>	A template for generating code to CRUD user interface
	<i>MenusView Template</i>	A template for generating for menu web application
	<i>TreeListForm Template</i>	A template for generating code which related to user interface with two options ; Tree of data view on the right, List of data view and Form on the left side
	<i>ListFormTemplate</i>	A template for generating code which related to user interface with Form
MVC_ Controller	<i>Controller Template</i>	A template for generating code to controller

Table 4.2 DDL Templates

Template	Name	Description
DDL	<i>DDLTemplate</i>	A template for generating DDL scripts to create the tables
	<i>Uml2services</i>	A template contains functions used in DDLTemplate

Table 4.3 WSDL Templates

Template	Name	Description
WSDL	<i>WSDLTemplate</i>	A template for generating WSDL files
	<i>WSDLServer</i>	A template for generating PHP service (server)

Two of these MVC_Model templates are the CRUDModelTemplate and WsModelTemplate. CRUDModelTemplate is a template used to generate code which related to CRUD (Create, Read, Update and Delete) database operation while WsModelTemplate can be used for generating code associated to web service functions. Figure 4.7 shows a template used for the MVC_Model : CRUDModelTemplate .

```

[module CrudModelTemplate('http://www.eclipse.org/uml2/3.0.0/UML')]
[template public CrudModelTemplate(aClass : Class)]
[file (aClass.name.concat('Model').concat('.php').toUpperFirst(), false, 'UTF-8')]

<?php
.....
class [aClass.name.toUpperFirst().concat('Model')] extends CI_Model {
    [for (p: Property | aClass.attribute)]
        var ${p.name}/;
    [/for]
    .....

    [for (o: Operation | aClass.ownedOperation) separator('\n')]
        function [o.name/]( $obj = null ) {
            [if (o.name.equalsIgnoreCase('save'))]
                $data = null;
                [for (p: Property | aClass.attribute)]
                    [if (p.name = 'id')]
                        $data->id = $this->db->insert_id();
                    [else]
                        $data->[p.name/] = $obj->[p.name/];
                    [/if]
                [/for]
                $this->return = $this->db->insert("$this->table_name",$data);
            [elseif (o.name.equalsIgnoreCase('update'))]
                $data = null;
                [for (p: Property | aClass.attribute)]
                    $data->[p.name/] = $obj->[p.name/];
                [/for]
                $this->db->where('id', $data->id);
                $this->return = $this->db->update("$this->table_name", $data);
            [elseif (o.name.equalsIgnoreCase('delete'))]
                $this->db->where('id', $obj->id);
                $this->return = $this->db->delete("$this->table_name");
            [elseif (o.name.equalsIgnoreCase('getbyId'))]
                $this->return = $this->db->query("SELECT * FROM ".$this->table_name."
                WHERE id = $obj->id")->result_array();
                $this->return = $this->return[ '[' / ]0[ ' ' / ];
            .....
        }
    [/for]
}
?>
[/file]
[/template]

```

Figure 4.7 An excerpt of one template for the MVC_Model : CrudModelTemplate

Transformation structure in Aceleo consists of module and template.

[module]

[template]

Module defines reusable transformation libraries and template defines transformation itself.

[module CrudModelTemplate('http://www.eclipse.org/uml2/3.0.0/UML')] shows input meta-model URL. Template structure starts with **[template public CrudModelTemplate(aClass : Class)]** and end with **[/template]**.

Template will generate file xxxModel.php with :

[file (aClass.name.concat('Model').concat('.php').toUpperFirst(), false, 'UTF-8')]

Class xxxModel will extends class CI_Model and property of class will be iterated to be variables with :

```
class [aClass.name.toUpperFirst().concat('Model')] extends CI_Model {  
    [for (p: Property | aClass.attribute)]  
        var $[p.name/];  
    [/for]
```

The **for** block is used for specifying a text segment that needs to be processed repeatedly over a set of model elements. This for block checks method of class then it will be iterated to be functions :

```
[for (o: Operation | aClass.ownedOperation) separator('\n')]  
    function [o.name/]($obj = null) {
```

The names of the functions in the class will be selected to generate all inside a **for** block code into the similar function in code. For example, if the name of the function is **save**, the block **save** will be generated in the function **save**.

```
    [if (o.name.equalsIgnoreCase('save'))]  
        $data = null;  
        [for (p: Property | aClass.attribute)]  
            [if (p.name = 'id')]  
                $data->id = $this->db->insert_id();  
            [else]  
                $data->[p.name/] = $obj->[p.name/];  
            [/if]  
        [/for]  
  
    $this->return = $this->db->insert("$this->table_name",$data);
```

Similarly, with the function of **update**, **delete**, **getById**, **getAll**, **getByLimit** and **filter**, the block of **update**, **delete**, **getById**, **getAll**, **getByLimit** and **filter** will be generated to the function of **update**, **delete**, **getById**, **getAll**, **getByLimit** and **filter** respectively.

In this case, MainCRUDGenerator.java is generated class and MainCRUDGenerator.mtl is a template file but both having same name. Figure 4.8 shows an excerpt of MainCRUDGenerator.java.

```
package id.ac.unila.simpel.main;
import java.io.File;

/**
 * Entry point of the 'MainCrudGenerator' generation module.
 * @generated
 */
public class MainCrudGenerator extends AbstractAcceleoGenerator {
    /**
     * The name of the module.
     * @generated
     */
    public static final String MODULE_FILE_NAME = "/id/ac/unila/simpel/main/MainCrudGenerator";

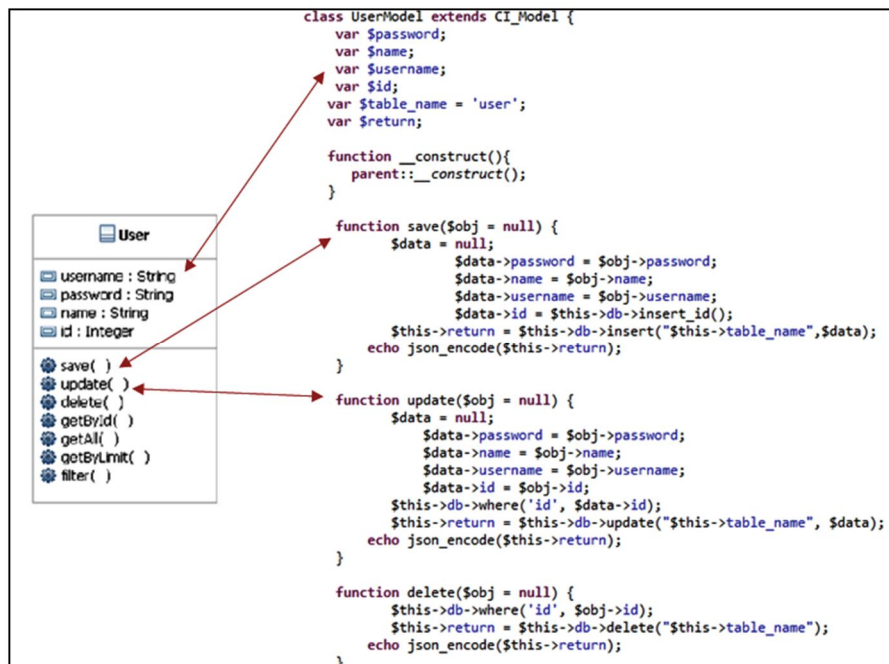
    /**
     * The name of the templates that are to be generated.
     * @generated
     */
    public static final String[] TEMPLATE_NAMES = { "MainCrudGeneratorClass" };

    /**
     * The list of properties files from the launch parameters (Launch configuration).
     * @generated
     */
    private List<String> propertiesFiles = new ArrayList<String>();

    /**
     * Allows the public constructor to be used. Note that a generator created
     * this way cannot be used to launch generations before one of
     * {@link #initialize(EObject, File, List)} or
     * {@link #initialize(URI, File, List)} is called.
     */
}
```

Figure 4.8 An excerpt of MainCRUDGenerator.java.

Figure 4.9 shows an example of source code of UserModel.php obtained from a User class by using CRUDModelTemplate file. From all CRUD models used in this case, it can be concluded that the generated code of CRUD models is not required customize code greatly due to the implementation has been defined on the template.



```
class UserModel extends Cf_Model {
    var $password;
    var $name;
    var $username;
    var $id;
    var $table_name = 'user';
    var $return;

    function __construct(){
        parent::__construct();
    }

    function save($obj = null) {
        $data = null;
        $data->password = $obj->password;
        $data->name = $obj->name;
        $data->username = $obj->username;
        $data->id = $this->db->insert_id();
        $this->return = $this->db->insert("$this->table_name", $data);
        echo json_encode($this->return);
    }

    function update($obj = null) {
        $data = null;
        $data->password = $obj->password;
        $data->name = $obj->name;
        $data->username = $obj->username;
        $data->id = $obj->id;
        $this->db->where('id', $data->id);
        $this->return = $this->db->update("$this->table_name", $data);
        echo json_encode($this->return);
    }

    function delete($obj = null) {
        $this->db->where('id', $obj->id);
        $this->return = $this->db->delete("$this->table_name");
        echo json_encode($this->return);
    }
}
```

Diagram illustrating the mapping between the User class and the UserModel.php code:

- User Class Attributes:**
 - username : String
 - password : String
 - name : String
 - id : Integer
- User Class Methods:**
 - save()
 - update()
 - delete()
 - getById()
 - getAll()
 - getByLimit()
 - filter()
- UserModel.php Code:**
 - Class definition: `class UserModel extends Cf_Model {`
 - Attributes: `var $password;`, `var $name;`, `var $username;`, `var $id;`, `var $table_name = 'user';`, `var $return;`
 - Constructor: `function __construct(){ parent::__construct(); }`
 - Save method: `function save($obj = null) { ... }`
 - Update method: `function update($obj = null) { ... }`
 - Delete method: `function delete($obj = null) { ... }`
 - Class closure: `}`

Figure 4.9 An excerpt of UserModel.php

The defined DDLTemplate is DDLTemplate.mtl. This template will create the DDL table elements from a model. This template will generate the files according to the model of class diagram. Figure 4.10 shows the DDLTemplate.

```
[module DDLTemplate('http://www.eclipse.org/uml2/3.0.0/UML')]
[import uml2services /]

[template public DDLTemplate(aModel : Model)]

[file (aModel.name.concat('.sql'), false, 'UTF-8')]
```

All the classes of the model will be in the iteration and the association of classes will be checked. Similarly, all the properties of classes will be in the iteration. The name of the table in database will be the same with the class diagram.

```
CREATE DATABASE IF NOT EXISTS `[aModel.name/]`;
USE `[aModel.name/]`;
[for (aClass:Class|aModel.allOwnedElements()->filter(Class))]
```

If a class is without the association, there is therefore no relation among database tables and as a consequence database can directly be created and field tables can directly be generated.

```
[if ( aClass.getAssociations()->size() < 1)]
    DROP TABLE IF EXISTS `[aClass.name.toLower()]/`;

    CREATE TABLE `[aClass.name.toLower()]/` (
```

If an association exists, the inter-table relations will first be examined on whether it is a one-to-many or many-to-many relationship.

```
[else]
[for (a : Association | aClass.getAssociations())]

[let notNavigable: Property = a.getFirstNotNavigableMemberEnd()]
[comment one to many /]
[if not (notNavigable.isMultivalued())]
DROP TABLE IF EXISTS `[aClass.name.toLower()]/`;

    CREATE TABLE `[aClass.name.toLower()]/` (
```

This also applies to joint table and inheritance, after which the database and field table can be generated.

```

[comment encoding = UTF-8 /]
[module DDLTemplate('http://www.eclipse.org/uml2/3.0.0/UML')]
[import uml2services /]

[template public DDLMySQLTemplate(aModel : Model)]

[file (aModel.name.concat('.sql'), false, 'UTF-8')]

.....
CREATE DATABASE IF NOT EXISTS `[aModel.name/]`;
USE `[aModel.name/]`;
[for (aClass:Class|aModel.allOwnedElements()->filter(Class))]
[if ( aClass.getAssociations()->size() < 1)]
    DROP TABLE IF EXISTS `[aClass.name.toLower()/]`;
    CREATE TABLE `[aClass.name.toLower()/]` (
[for (prop:Property|aClass.ownedAttribute)]
[if (prop.name.equalsIgnoreCase('Id'))]
    `[prop.name.toLower()/]` int(11) NOT NULL AUTO_INCREMENT,
[elseif (prop.type.name.equalsIgnoreCase('Integer'))]
    `[prop.name.toLower()/]` int(11) [if prop.getDefault() <> null]NOT [if] NULL,
[elseif (prop.type.name.equalsIgnoreCase('Boolean'))]
    `[prop.name.toLower()/]` bool [if prop.getDefault() <> null]NOT [if] NULL,
[elseif (prop.type.name.equalsIgnoreCase('String'))]
    `[prop.name.toLower()/]` varchar(255) [if prop.getDefault() <> null]NOT [if] NULL,
[elseif (prop.type.name.equalsIgnoreCase('BigInteger'))]
    `[prop.name.toLower()/]` bigint [if prop.getDefault() <> null]NOT [if] NULL,
[elseif (prop.type.name.equalsIgnoreCase('Edate'))]
    `[prop.name.toLower()/]` date [if prop.getDefault() <> null]NOT [if] NULL,
[if]
[/if]
    PRIMARY KEY (`id`)
    ) ENGINE=INNODB AUTO_INCREMENT=0 DEFAULT CHARSET=utf8;
[else]
[for (a : Association | aClass.getAssociations())]
.....

```

Figure 4.10 An excerpt of DDLTemplate

On the other hand, Figure 4.11 shows the WSDLTemplate. This template converts a simple representation of WSDL interface into the required WSDL files. It provides the information necessary to invoke a web service by creating WSDL file, an XML document that describes the purpose, location, and information on how to access a web service. The WSDL file also describes the methods and their data types. Figure 4.12 shows an example the generated WSDL file from model.

```

...
<message name="[aClass.name/]_serviceRequest">
  [for (p: Property | aClass.attribute)]
    <part name="[p.name/]" type="xsd:[p.type.name/]" />
  [/for]
</message>

<message name="[aClass.name/]_serviceResponse">
  [for (p: Property | aClass.attribute)]
    <part name="[p.name/]" type="xsd:[p.type.name/]" />
  [/for]
</message>

<portType name="[aClass.name/]PortType">
  <operation name="[aClass.name/]">
    <documentation>[aClass.name/] - docs</documentation>
    <input message="tns:[aClass.name/]_serviceRequest"/>
    <output message="tns:[aClass.name/]_serviceResponse"/>
  </operation>
</portType>

<binding name="[aClass.name/]Binding" type="tns:[aClass.name/]PortType">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="[aClass.name/]">
    <soap:operation soapAction="urn:[aClass.name/]#[aClass.name/]" style="rpc"/>
    <input>
      <soap:body use="encoded" namespace="urn:[aClass.name/]"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    <output>
      <soap:body use="encoded" namespace="urn:[aClass.name/]"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
  </operation>
</binding>
...

```

Figure 4.11 An excerpt of WSDLTemplate

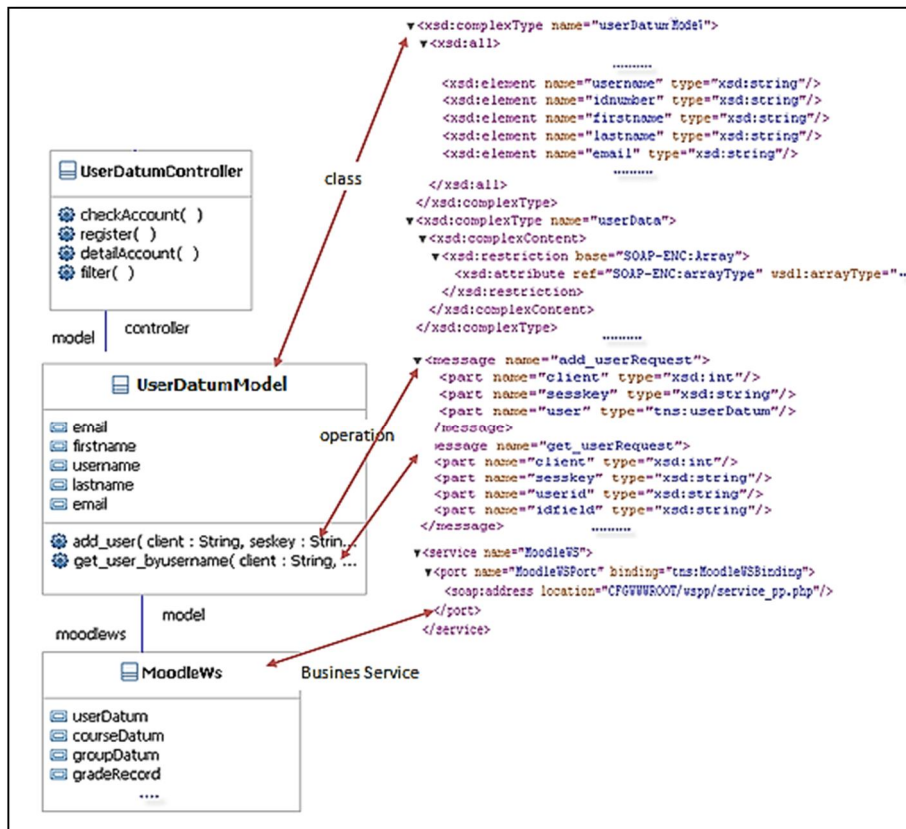


Figure 4.12 An excerpt of UserDatum.wsdl

4.3.2.3 Code Generator

Using code generator significantly reduces time required for web development. The goal of code generator is to generate repetitive source code while increasing the quality of source code by generating standardized code. It also decreases number of errors and repetitive tasks which are common in manual coding. The generator uses specific inputs data model and templates then produces output by applying defined rule. Outputs can be different kinds of text files such as PHP files, Java source code files, SQL scripts and XML.

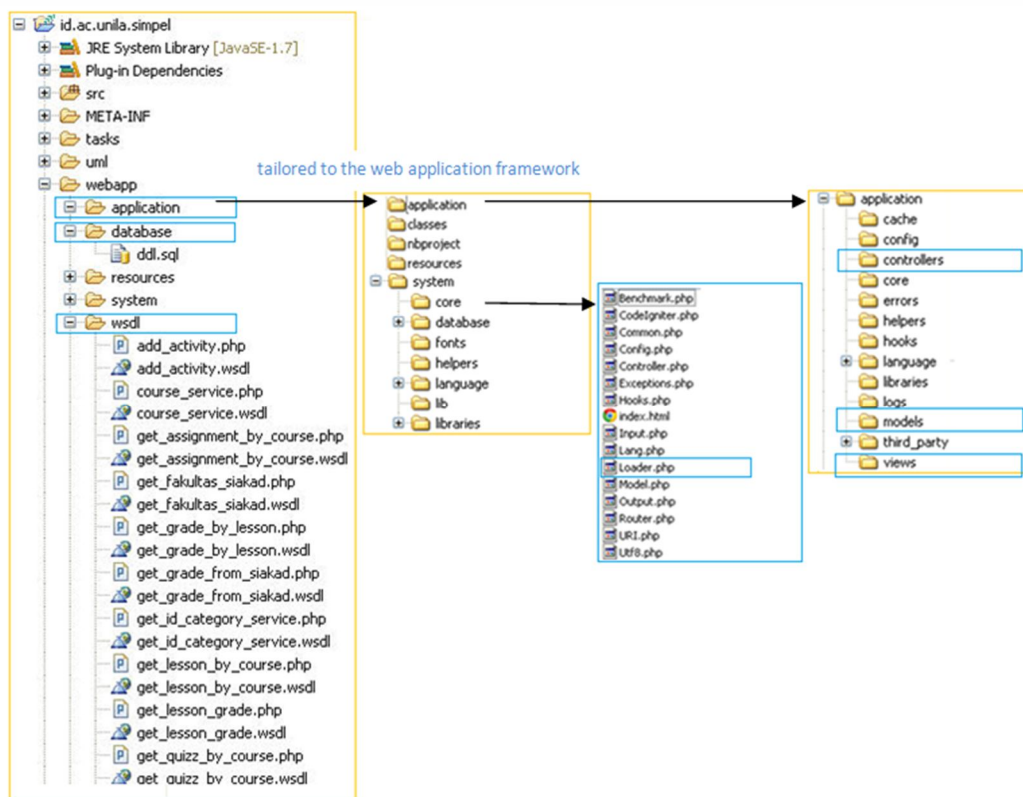


Figure 4.13 Structure folders and files generated by the generator code

Figure 4. 13 shows the structure folders and files of a web application interface generated by the generator code. This structure is divided in three main folders : Application, Database and WSDL. The last one contains the WSDL files and PHP files to define the interface to web services. All the generated WSDL dan PHP files in this folder are then configured by the related server. The folder Database contains the DDL SQL script that allow us to create the data base

required for the web application. All the generated DDL files in this folder are then executed by database server. The folder Application contains the folder of the Models, Controllers and Views. For the implementation all the folders and files are tailored to the web framework Codeigniter. All the files in Model, View, Controller are set by file Loader.php in system/core.

4.3.2.4 Implemented Web Application and Framework

The result of generated and handwritten code are tailored to the web application framework for system implementation. Using a web application framework or not is a decision that should be taken by developers and project managers in developing web application. Therefore the advantages and disadvantages of using a web application framework should be considered before starting the development. Usage web framework has advantages in many ways such as for developers, they do not have to deal with many common programming tasks. Web application framework will reduces time consumed on code maintenance and future development. Code maintenance is considered by most developers as complex, technically difficult and tiresome task. It is also facilitate other developers to continue development of the web applications without having to spend time learning the code. There are also various of libraries and helpers included in the web application framework. The same function can usually be accomplished with less code making it easier to maintain. Likewise for some features will be added in the upcoming development. On the other hand, the disadvantages of using a web application framework that the complexity and overhead of the framework code can reduce application performance and creating a greater problem for the essential hardware. Tight conventions constrain the flexibility and creativity of developers when developing the web application. In view of this, the advantages can be obtained much more than disadvantages when web application is developed using an appropriate framework.

There are literally numerous of web application frameworks, but a slight number stand out in particular. It because of their widespread popularity and usage the important Model-View-Controller pattern. A number of these can be classed as fully MVC-compliant web application frameworks. Some of the better known web application frameworks such as Ruby on Rails (Ruby), CodeIgniter (PHP), Symphony (PHP), CakePHP (PHP), Zope (Python), DotNetNuke (VB.NET), Apache Struts (Java), Spring (Java). The development of web application system in

this work is combined with a web application framework CodeIgniter, leading to faster PHP web application development. This framework is designed to simplify programming and encourage code reuse by setting forth folder structure, documentation, guidelines and libraries. It is also expressed by the directory structure. The selected framework then adapted to the needs of the web application interface. Based on the framework, there are 3 types of main program to be used: the model, view and controller. Models used to access data from persistent media. View is used for the presentation of data to users. Controller is used to connect each component in the application. There are also support files that shown in figure 4.14. This figure shows the MVC pattern that used to illustrate the structure of web application.

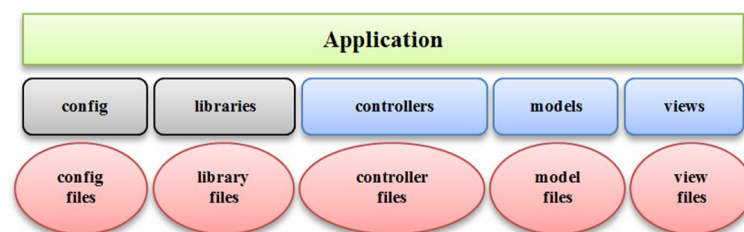


Figure 4.14 MVC pattern of web application

According to the files shown in Figure 4.13, the architecture of the web application interface is can be seen in see Figure 4.15. The Web application interface that is accessed from a web browser is implemented by means of HTML code. This code is produced by the PHP files included in the folder of Views, Controllers and Model.

The produced HTML code structures the information that must be provided to users without considering aspects related to images or aesthetic properties. These aspects are considered by introducing into the HTML code references to the files included in the folders CSS, Javascript dan Images. To display the information, the View is rendered by the Controller. Before rendering the View, the Controller create an object from the Model and also call functions from the Model. To obtain the data from one server which provide services, the Model call services from the server providers. After obtaining the corresponding data, the Model send the data to the Controller which then render it to the view. The Model is also used for connecting to the database in order to get and save the data. Folder Config contains files related to the

configuration of the database. All the files in Model, View, Controller and the Config are set by file Loader.php.

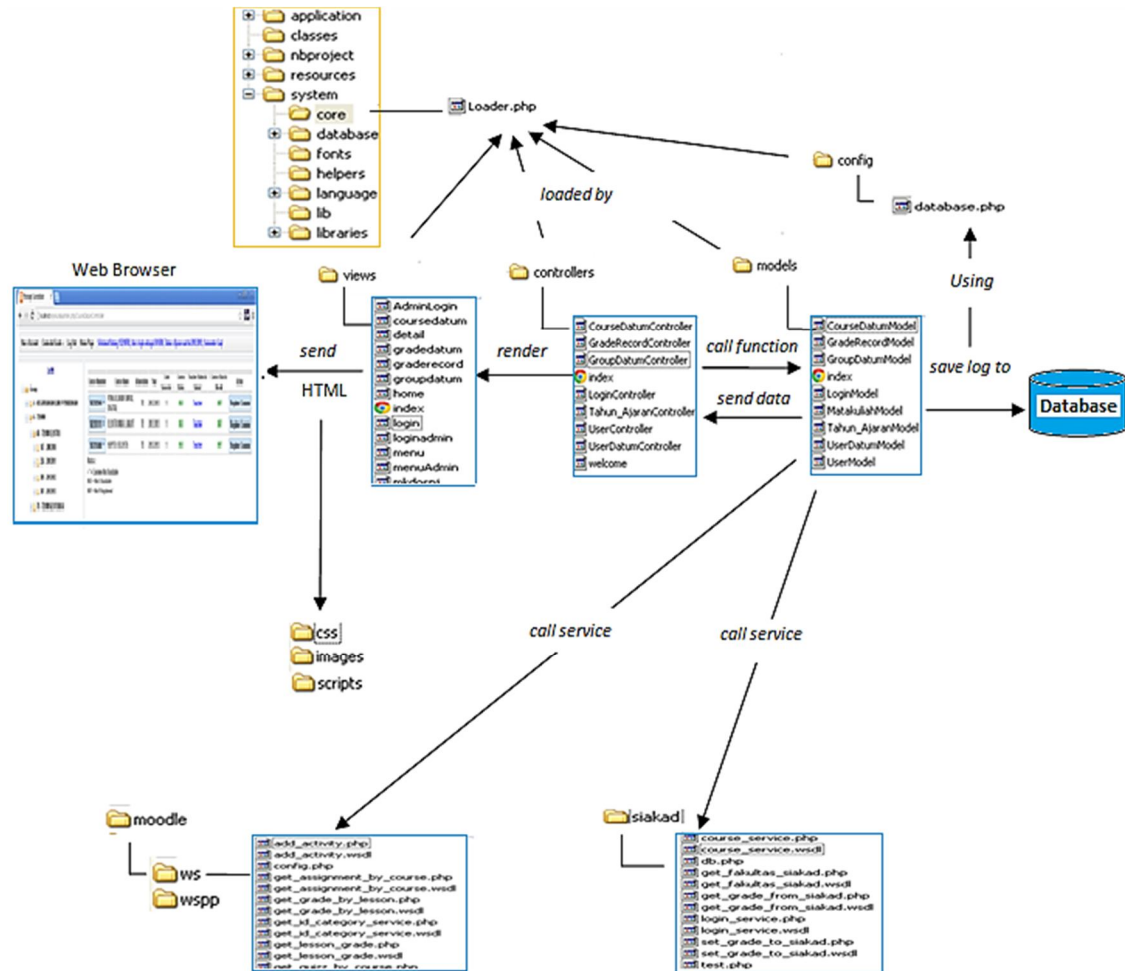


Figure 4.15 Architecture of the generated web application

4.3.3 Guideline for WIS development process

To guide developer to develop WIS by using our method, we provide steps of development process that can be followed. This guideline focuses on activities related to the development of WIS based on analysis results conducted in the previous chapter. Each step shall generate a model with explicit needs, and as a consequence after a model has been created in a given step, efforts can focus on model development for the subsequent step. It also added the problem identification as the a first step. The steps illustrated as follow :

Step 1. Problem Identification

The initial stage is to identify problems confronting educational institutions, in this case universities. This stage is intended to ensure that the appropriate solution effectively deals with the problem. As such, the development of IS can successfully address emerging problems. This activity begins with the analysis of the institution's situation, particularly in terms of its business process.

Step 2. Requirements Specification

This step should initially begin with the identification of business processes and system capabilities of each existing system for integration purposes. Redundant processes and information should be discharged to avoid the duplication of input data, divergence in information flows or inconsistent processes. This step illustrates the institution's business process after the solution that has been analyzed is applied for the existing problem. The business process can describe the environment and needs specific to the IS to be developed.

Step 3. Define Data and Web Services

The following step is data and information mapping from available documentation to determine data variety and potential data combinations. This generally involves the collection of detailed information, i.e., name, format, attribute, sources, level of security, life time and users of data. Web services can then be defined or selected from the list of web services on existing systems. The expected output is the mapping of data and service catalogue containing the list of web services needed from the existing systems.

Step 4. Design Models

This step is aimed at the development of PIM that focuses on systems operation that does not rely on a specific platform. In this step, a set of models is designed related to the web application interface, database and web services. In our proposed method, we have created metamodel to design the UML models and templates for transformation into source codes. This is necessary to support WIS development according to the proposed method in a comprehensive manner consistent with the principle of MDA SOA.

Step 5. Generate Source Codes

This step concentrates on the development of PSM, and the required platform is a web-based application that suits WIS needs. The usefulness of the templates is to transform models designed in step.4 to source codes. By using the models and our defined templates, the code generator will create all application source code based on MVC pattern (PHP files), DDL (SQL files) and WSDL files. All three types output are created through a transformation process from the PIM produced. MVC web application is drawn from the application and information aspect, while WSDL is obtained from the service aspect and DDL from the data aspect.

Step 6. Complete and Customize the Codes

Handwritten (manual) codes need to be added after generation for complex business logic and graphical user interface (GUI) parts.

Step 7. Finalize the Process

The result of generated and handwritten code are tailored to the framework for system implementation. Then to finalize the process development the web application has to be configured in server.

Table 4.4 shows the steps that must be followed with regard to the activity and output or models to the development of WIS.

Table 4.4 Activity and Output of WIS development process

Step	Activity	Output
Problem Identification	Identify existing problems in the educational institutions	HEI Business Function
		Institution Business Process
		Problem Analysis
		Solution Analysis
Requirements Specification	Identify business processes and system capabilities of each existing systems. This step is associated with an enterprise architecture particularly business architecture	Business Process Analysis
		Functional Requirements
		Integration Requirements

Define Data and Web services	Define data and web services needed. This step is associated with an enterprise architecture particularly information system architecture	Data Mapping
		Service Catalogue
Design Models	Design models for web application, database and web services needed	MVC models
		DDL models
		WSDL models
Generate Source Codes	Use MVC Templates	Generated web application source codes (PHP files)
	Use DDL Templates	Generated SQL files
	Use WSDL Templates	Generated WSDL and PHP files
Complete and Customize the Codes	Manual coding	Modified source codes
Finalize the Process	Tailored to web Framework	Ready web application system

4.4 Chapter Summary

In this chapter we proposed method based on MDA-SOA for developing and integrating WIS in the educational institutions. The WIS development method comprises five main steps : creating business models, creating process models, defining templates, code generator and web application source code. To guide developer to develop WIS by using our method, we provide steps that can be followed in development process. Each step shall generate a model with explicit needs, and as a consequence after a model has been created in a given step, efforts can focus on model development for the subsequent step. Problem Identification, Requirements Specification, Define Data and Web Services, Design Models, Generate Source Codes, Complete and Customize the Codes, and Finalize the Process.

The contribution of this chapter is two-fold. First, the proposed method provides a systematic procedure for developing and integrating existing web information systems. We defined web development process including CIM modeling, PIM modeling, and PIM to PSM and to web source code transformations. These results provide a greater insight for understanding and reducing the gap among PIM, PSM and code. Second, the proposed method provides templates to automate the transformation among PIM to code and the communication bridge between web application code to efficiently develop and increase productivity for WIS development.

Chapter 5 SIMPEL : A Case Study

This chapter presents a development of WIS by applying the proposed method in previous chapter to integrate two existing web information systems. A case study conducted by designing the SIMPEL as an web interface between academic information systems and Course Management System at University of Lampung (Unila, Indonesia). It also describes the functionality, service categories, the generated web application of SIMPEL and completed by demonstration scenario. The implemented SIMPEL is then analyzed in the teaching/learning activities at Department of Electrical Engineering Unila.

5.1 Chapter Introduction

Easy access to accurate information is required to provide timely information to assist the management of University of Lampung (Unila). Unfortunately, current systems in the Unila are characterized by a large degree of overlap in data elements, inconsistent data, heavy dependency on manual process, and difficulty in sharing and reporting of information. Based on the implementation experience using the information systems the integration of the system is required in order to enhance the effectiveness and efficiency. It is more serviceable if some data which were previously entered in one system is also available in other systems automatically. The actions initiated in one system will also affect others so the users should not make a transaction twice while using them. Also, the decision making is expected to be more accurate and timely because information can be obtained immediately [78]. This can give positive impacts to the university's efforts to provide the highest levels of service to students, teachers, and other constituents.

This chapter illustrates functionalities derived from the integration of two existing systems established in a university developed based on the MDA and SOA approaches that previously described on chapter 4. We discussed a web information system that serves as a web application interface using the web service technology to provide a level of interoperability among existing applications. The web application interface (called SIMPEL) as an interface between two existing applications in Unila. The web information systems are integrated to share data as well as providing an additional functionality which not yet covers in the existing systems to enrich their utility. This web application interface is capable to make one system use the most of input from other systems to provide the new information.

The SIMPEL benefit is creating synchronization between academic data and teaching/learning resource. It also improves several functionalities which are lacking in the MOODLE. This work shows how the proposed method mentioned in previous chapter helps to develop WIS that invokes web service to enable the interoperability of two existing systems.

5.2 Existing Web Information Systems

The need to integrate existing information systems in HEI arises from the necessity to combine data held by departments, faculties or other work divisions in order to allow data access for different purposes. Through the integration of different systems, data previously entered in one system is also automatically available in other systems. Users need no longer make a transaction twice when using the integrated system because the actions initiated in one system will also directly affect others. The Unila has similarly shown interest in developing an integrated information system due to the number of existing web information systems with different platforms and databases. Information systems in Unila were developed by using own source-based development and open source-based development, including several special purpose applications developed by the government and those ready to be implemented.

Concerning WIS development, based on Unila's needs assessment, the Academic management and Knowledge Management domain shall be given the highest development priority. The following example focuses on the development of a web information system to integrate the existing systems in Academic Management and Knowledge Management as a solutions and

strategies for bridging the gap mentioned in section 3.8. As an illustration about existing related web information systems in the two domains is shown in Figure 5.1.

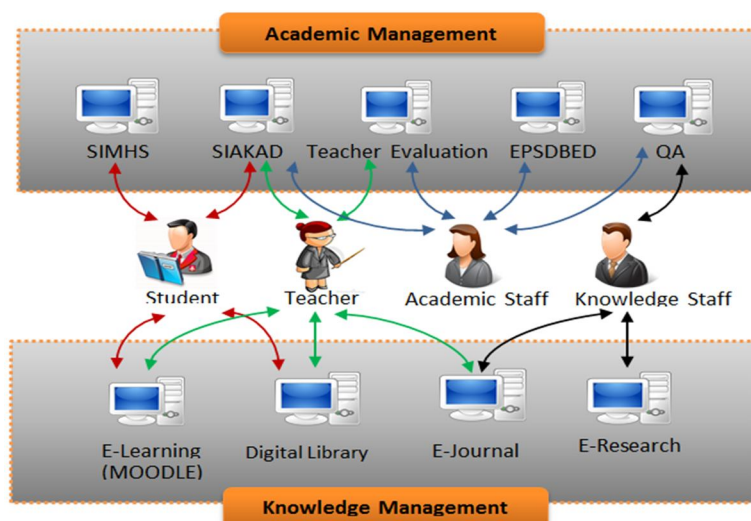


Figure 5.1 Existing WIS in Academic and Knowledge Management

5.2.1 Course Management System

Unila currently has two running web information systems concerning to the academic and teaching/learning process, i.e. the MOODLE and SIAKAD⁵. MOODLE stands for Modular Object-Oriented Dynamic Learning Environment is an example of a Course Management Systems (CMS) [79]. MOODLE is one of the most popular open-source CMS, with over 80.000 registered instances, and well over 71 million users [80]. It is a free web application to create an e-learning system to enrich the academic activities in teaching/learning processes while students are off the class, and also provides necessary support to the practical in laboratory work [81]. MOODLE users are largely students, teachers, administrators and visitors. As supplement for traditional learning MOODLE is very beneficial in order to support conventional learning process by giving an opportunity to students to learn independently or in a group.

MOODLE allows users to post journals and resources, submit and collect assignments, engage in dialogues and group discussions, project work, assessments and course evaluation. The

⁵ SIAKAD stands for Sistem Informasi Akademik. This is Indonesian terminology meaning Academic Information System.

MOODLE have facilitated the development of web services so that a customer can select the most appropriate available service or add the new service as needed. The screenshot of a web page Unila's MOODLE is shown in Figure 5.2.

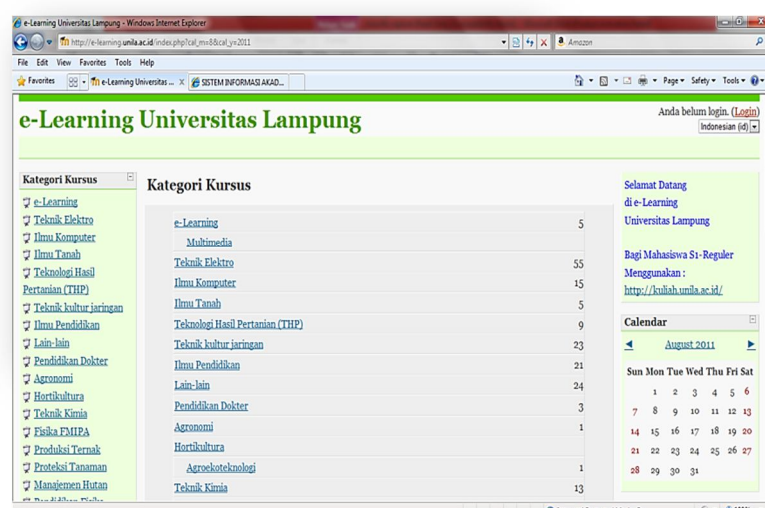


Figure 5.2 A Screenshot of MOODLE Unila

5.2.2 Academic Information System

The SIAKAD (Academic Information System Unila) meanwhile was developed by Unila as a central academic information system specifically for the purpose of managing information related to the university's existing academic programs. The screenshot of a web page Unila's SIAKAD is shown in Figure 5.3. The SIAKAD web application handles all academic and business processes within the university. It is equipped with supporting features that include data management on students and teachers, courses, curriculum, course schedule, study plan, list of attendees, room allocation, study results, grade transcripts and status of student tuition payment. The SIAKAD has facilitated the administrative work turn to become faster and easier compared to the situation before it was the launched. The system allows students to fill out their study plan, and the teacher advisor may give their advice and validity to the student's study plan while they are off the campus. Moreover, as soon as the responsible teachers of the course input the course grades to the student's record after the final exam, the students will find out their study result. This system provides a facility for study programs and faculties to

manage useful report regarding the student's academic record. The challenge is the SIAKAD has never been integrated into other systems before. It does not have a web service hence for implementing this approach, the web services to the SIAKAD that allows other systems to access its data has to be created first. We provided SIAKAD web services and discussed in our previous work.

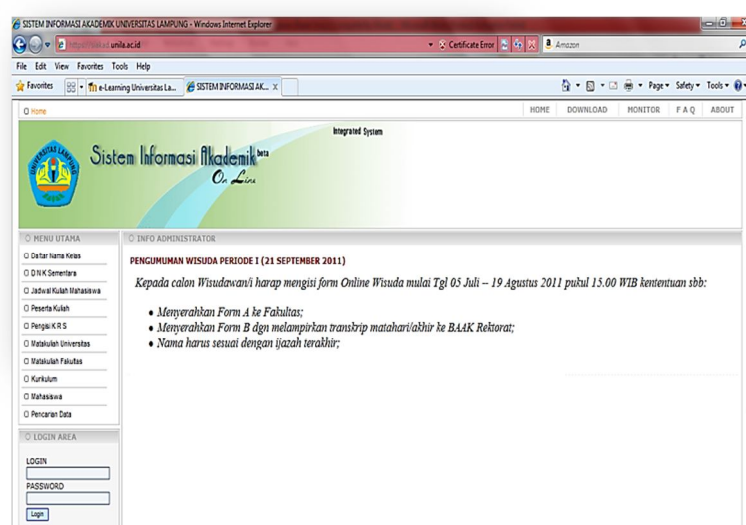


Figure 5.3 A screenshot of SIAKAD (Academic Information System) Unila

5.3 The SIMPEL Development

5.3.1 Description

The MOODLE and SIAKAD have common sets of data, but interaction is not able between them. By synchronizing their operations, teaching/learning activities managed by MOODLE displayed based on the current faculty schedule that already fixed on the SIAKAD in each semester. Likewise, available e-learning courses are synchronized with the academic curriculum and student study plans. Towards the end of each semester, students' grades earned from MOODLE online tests are calculated along with offline test grades in the physical classroom to be submitted directly to SIAKAD as their final study grades. To achieve this, the integration of

SIKAD and MOODLE is essential through a web application interface known as SIMPEL⁶. An illustration of the SIMPEL is shown in Figure 5.4.

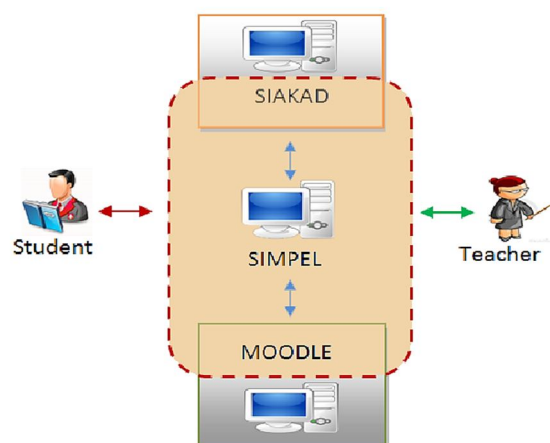


Figure 5.4 Overview SIMPEL

SIKAD and MOODLE have major functions that should be accessible to users for academic and teaching/learning purposes. For several other functions that users may require, both WISs have limitations that hamper their ability to address user needs. Integration through a web application interface can therefore be a workable solution. The integration process may cover data and process integration in SIKAD and MOODLE. The rationale for data and process integration is provided in table 5.1.

Table 5.1 Integration parts and reasons for integration

Integration of		Description	Reasons for Integration
Process	Data		
User Management	user	SIKAD and MOODLE have the same data of users – students and teachers. Users are registered and managed by the respective systems.	Through integration, with regard to data users of SIKAD and MOODLE, the MOODLE administrator no longer need to input data one after the other or upload a user registration data file. Users will only require a quick registration or activation their of MOODLE account.

⁶ SIMPEL is Indonesian terminology, stands for Sistem Informasi Pembelajaran E-learning. In this case the term is used for a web application that serve as an interface (called web application interface).

Course Management	Course, Curriculum, Faculty, Department, Study_program	Teachers manage courses through MOODLE by adding new courses, and editing and deleting available courses manually.	Through integration, course data available at SIAKAD and MOODLE is created automatically according to course data at the faculty, department and study program. The automatic course coordinators are teachers who also conduct face-to-face classes. Course data in MOODLE is also presented according to the schedule for the ongoing semester, synchronized with the schedule data at SIAKAD. Teachers therefore need not retrieve course data in advance in order to access and register courses under their responsibility for the ongoing semester.
Course Activity	Schedule, DNK, Year academic	Teachers can complement courses with various activities such as quiz, survey, forum, training material, assignment, chat, Wiki and others.	Through integration, with regard to schedule data and academic year, teachers will not only be able to add new activities, but also modify course activities for the previous semester for re-registration in the ongoing semester. Data integration also allows students to only view and enroll courses suitable with their study plan for the ongoing semester which prevents any mistake in course enrolment that may not be suitable with the study plan.
Group	User, Grade	Once activities and tests are completed at MOODLE, students can obtain their grades.	Through integration, students' grades shall become the basis for classifying them into groups. These groups are created by the system based on the score range of the test. Each group shall receive different activities and actions determined by the teacher (this is known as the learning strategy).
Grade Calculation	User, Final grade, Study result, DAK	By the end of the semester, students' grades obtained through activities conducted through MOODLE are calculated by including grades for offline activities	Through integration, grade data can be calculated at MOODLE and the final grade can be sent and stored automatically through SIAKAD.

		(face-to-face classes). Teachers calculate the final grade before inputting it into SIAKAD manually.	
--	--	--	--

Figure 5.5 shows in more detail form, the data flow as well as the interactions of users with system. The SIMPEL oriented to control MOODLE usage, monitoring user's activities as well as improving several functionalities which are lacking in the MOODLE. Those new functionalities are presented in the form of sequence of activities, start from the student's study plan and course schedule as inputs from SIAKAD until the student's learning outcome assessment.

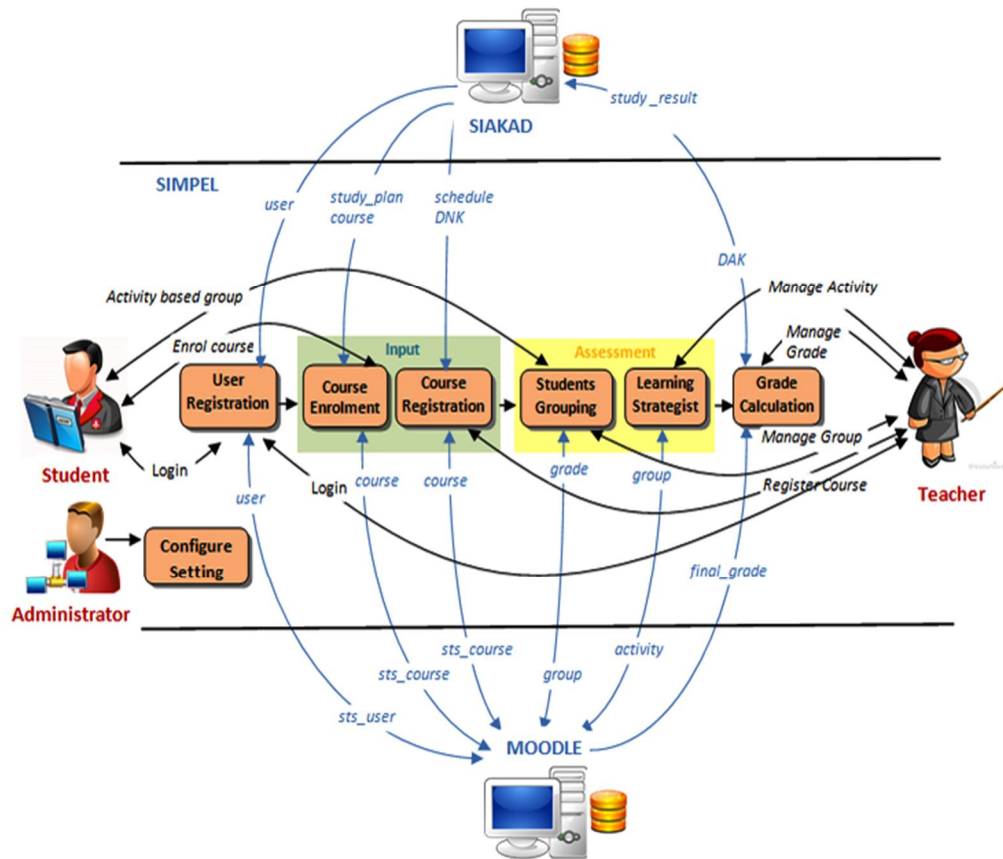


Figure 5.5 Data flow and user interaction with SIMPEL

It also provides the facility to design different activities to different group of students. In the assessment process, the system will inform the teacher the progress of the students based on

their online test results in MOODLE. Additionally, the system will guide to group the students based on their ability automatically and advise appropriate learning strategies to them. Learning strategies are determined based on differentiation on student ability. They are implemented through different learning activities, assignment, support, and time allocation adjustment given for each learning material.

By using the SIMPEL application, students can register to the MOODLE directly. This process eliminates the need for the re-entry of student personal information as the required data is already available in SIAKAD. Students allow to enroll courses in the MOODLE and do learning activities based on their valid courses in study plan in the SIAKAD. Upon completion of course materials and doing online tests through MOODLE, students are automatically grouped based on the score range earned from these exams. These groups will have different activity at MOODLE based on learning strategies decided by teacher through SIMPEL.

The description of functionality each WIS (SIAKAD, MOODLE, SIMPEL) illustrated in tables bellow. It includes the constraint/dependency and benefit of integrating systems.

5.3.1.1 User Registration

Table 5.2 User Registration

SIAKAD	Student's account in the SIAKAD is directly added by the SIAKAD administrator based on the data of new students registration in Unila each year academic. Students do not need to register by their self.
	Teacher's account in the SIAKAD is directly added by the administrator based on the data of teachers/ new teachers each faculty. Teachers do not need to register by their self.
MOODLE	Students need to create new account by their self on the MOODLE by filling the form registration. An email confirmation will be sent to the email address that entered in the form. Moodle administrator set up a class and login for teachers so the teachers do not need to register by their self.
SIMPEL	Since students/teachers can access to the SIMPEL, they are allowed registering/activating their account in MOODLE. Students/teachers can access applications at the same time by using SIMPEL.
CONSTRAINT/DEPENDENCY	Students/teachers do the registration/activation to MOODLE applications through an application interface (SIMPEL). Students/teachers who are not registered in the SIMPEL can also access the MOODLE directly but cannot perform some activities for example

	students cannot access the full material due to the requirements of study plan in SIAKAD. On the other hand teachers cannot perform some activities for example teachers cannot use data assessment of a course from the MOODLE for scoring process.
INTEGRATED	Students/teachers do not need to re-entry their detailed personal information to register to other applications. In this case to register to the MOODLE. The data of students/teachers have already provided by the SIAKAD

5.3.1.2 Course Enrolment

Table 5.3 Course Enrolment

SIAKAD	No function Course Enrolment in SIAKAD
MOODLE	Students will see list of all the courses in their department although not all courses available in on-going semester. They will search manually the courses that they wish to enroll. To enroll the course students can click on the text of the course's title.
SIMPEL	Students can enroll courses based on their valid courses in study plan. The status of course 'registered or not registered' will be shown in each course in student's study plan. Students can enroll only the courses that have registered status (registered by the teachers).
CONSTRAINT/DEPENDENCY	In order to enroll a course there are two prerequisite; (1) the course has already filled in the student's study plan in the SIAKAD and (2) the course has already registered by the teacher in the MOODLE.
BENEFIT OF INTEGRATION	Students only see the course in their study plan. They do not need to see all the list of courses in their department. Data course in study plan provided by SIAKAD. Data registered course provided by MOODLE.

5.3.1.3 Course Registration

Table 5.4 Course Registration

SIAKAD	The data of courses each on-going semester stored in SIAKAD and can be used as a reference for teachers.
MOODLE	No function Course Registration in MOODLE
SIMPEL	The teachers as coordinator courses will see a list of their courses in the on-going semester. Teachers can register courses and provide e-learning activities in MOODLE so that students can enroll the courses.
CONSTRAINT/DEPENDENCY	In order to register courses there are two prerequisite; (1) the courses in teacher's list of courses (as a course's coordinator) in on-going semester and (2) the courses that have e-learning activities in MOODLE

BENEFIT OF INTEGRATION	Teachers only see the courses in their list of courses (as a course's coordinator). They do not need to see all the list of courses in their department. Teachers can choose which courses material to be used in on-going semester.
-------------------------------	---

5.3.1.4 Student Grouping

Table 5.5 Student Grouping

SIKAD	No function Student Grouping in SIKAD
MOODLE	There are students grouping facility in MOODLE, but it cannot automatically grouping the students based on their grade.
SIMPEL	The teachers as coordinator courses will see a list of their courses in the on-going semester. Teachers can select courses and group of the students based on students grade automatically.
CONSTRAINT/DEPENDENCY	In order to create a group of students, the student's status must be in teacher's list of number class in on-going semester.
BENEFIT OF INTEGRATION	Teachers can automatically grouping the students based on their grade.

5.3.1.5 Learning Strategist

Table 5.6 Learning Strategist

SIKAD	No function Learning Strategist in SIKAD
MOODLE	No function Learning Strategist in MOODLE
SIMPEL	The teachers as coordinator courses will see a list of their courses in the on-going semester. Teachers can select their courses and advise appropriate learning strategies to students. Learning strategies are determined based on differentiation on group of students.
CONSTRAINT/DEPENDENCY	In order to create a learning strategist for students there are two prerequisite; (1) only for students who in teacher's list of number class in on-going semester. (2) the student has already have a group.
BENEFIT OF INTEGRATION	Teachers can advise the appropriate learning strategies to students based on their group.

5.3.1.6 Grade Calculation

Table 5.7 Grade Calculation

SIKAD	No function Grade Calculation in SIKAD
MOODLE	No function Grade Calculation in SIKAD
SIMPEL	The teacher as coordinator courses will see a list of their courses in the on-going semester. Teacher can select their courses and calculate grades of students. Students' grades earned from MOODLE online tests are calculated with offline test grades in the face-to-face classroom.
CONSTRAINT/DEPENDENCY	Grade calculation only for grade of students who in teacher's list of number class (as a course's coordinator) in on-going semester.
BENEFIT OF INTEGRATION	Teachers can calculate the grades of students automatically by input each component assessment. Teachers can submit final grades of students directly to SIKAD after finish calculating.

5.3.2 Functionality

To show the functionality of the system, a use case diagram for users (students and teachers) as depicted in Figure 5.6 is created. Users use SIKAD account to login to SIMPEL. Only users authenticated by SIKAD can access SIMPEL and register to MOODLE. Students can enroll courses and do learning activities based on their valid courses in study plan. Study plan's students are maintained in SIKAD and their activities in a course are maintained in the MOODLE. After finishing specific material and doing online test at MOODLE, students are automatically grouped based on the score range of that test. These groups will have different activity at MOODLE based on learning strategies decided by teacher through SIMPEL.

Teachers are in charge of managing course, grade and group of students. A responsible teacher can register the course at SIMPEL. This registration will enable this course to be shown in MOODLE in the running semester. In other words, previously registered course will not be automatically registered in the running semester. Teacher then manages groups and learning activities based on grade of online test. Final grade of one course can be calculated from grade of online and offline activities and stored at SIKAD automatically.

View group member	Student	View group which already grouped by teacher based on student's grade
Register user	Student, Teacher	Registration is done by using SIAKAD's account. If user account is valid, user can register and have account as a user in SIMPEL
Perform activities in MOODLE	Student, Teacher	User can perform activities in MOODLE by selecting enrolled course (download material, Forum, Chat, Discussion, Project work, Assessment)
Check status in MOODLE	Teacher	Teacher can check the course's status
Manage course	Teacher	Teacher can manage the course
Register course	Teacher	Course can be registered only by a coordinator course
View course schedule	Teacher	View list of course schedule in accordance with the DNK (a list of number class that contain of students in one course)
Manage grade and course	Teacher	Teacher can manage grade and course
View lesson	Teacher	View lesson each course
Automatic grouping	Teacher	Teacher can automatically do students grouping
Update learning activity	Teacher	Teacher can update learning activity
Input grade in SIAKAD	Teacher	Input student's grade in DAK and save in SIAKAD.
Update course info	Teacher	Update information of course.
Calculate grade	Teacher	Accumulate student's score (activities in MOODLE and face-to-face classes).
View list of DAK	Teacher	View DAK (a list of class numbers that contain of students and grades).
View statistic	Teacher	Statistic each of department: list of course, number of course, number of student enrolled, user attendance, etc.).
Maintenance	Adminis-trator	Maintenance of course and user
Configuration	Adminis-trator	Setting the configuration

5.3.3 Services Categories

As an initial step for developing the integrated system is mapping the data and information from the documentation of existing application. It aims to determine the variety of data and also obtain the potential combinations of its. Generally it can collect the detailed information i.e. name, format, attribute, sources, level of security, life time and users of data. Then web services can be defined and selected from the list of web services on the existing systems. The output of this step is the map of data and the service catalog that contains the list of web

services needed from the existing systems. By identifying available web services in MOODLE and SIAKAD we described a list of service catalog as is shown in table 5.9. To use the MOODLE web service we apply the plug-in web service of the OK Tech MOODLE Web Service.

Table 5.9 Available Web Services

Business Service	Service Server	Service	Description
User registration	SIKAD	login_service	To login to SIMPEL
	MOODLE	get_user_byusername	To get user by user name
		update_user	To update user
		add_user	To add user
Course enrolment	SIKAD	course_service	To get List of courses
	MOODLE	enrol_students	To enrol student to course
		affect_user_to_course	To affect user to course
Course registration	MOODLE	add_teacher	To add teacher to course
		add_course	To add course
		affect_course_to_category	To affect to category
		affect_user_to_course	To affect user to course
Student grouping	MOODLE	get_course_byidnumber	To get course by id course
		get_groups_byname	To get group by group name
		add_group	To add group
		get_group_members	To add group member
		affect_user_to_group	To affect user/member to group
Configure Setting	MOODLE	add_category	To add category (faculty/department/year academic)
		get_categories	To get categories

Apart from the available web service, we have also created web services that are not available in SIAKAD or MOODLE required for the business process already defined. Table 5.10 presents a list of the web services created.

Table 5.10 Created Web services

Business Service	Service Server	Service	Description
Course enrolment	MOODLE	tree_list_course	To get list of courses and create tree
Course registration	MOODLE	get_id_category_service	To get id category by academic year and parent category
		tree_list_course	To get list of courses and create

			tree
Student grouping	MOODLE	get_grade_by_lesson	To get grade by Lesson ID
		get_lesson_grade	To get grade from each lesson
		get_quizz_by_course	To get quiz from each course
Learning strategist	MOODLE	view_group_by_lesson_id	To get group from each lesson
		view_lesson_by_course_id	To get lesson from each course
		add_activity	To add activity to group
		get_assignment_by_course	To get assignment by Course id
		get_lesson_by_course	To get lesson by course
Grade calculation	SIKAD	set_grade_to_siakad	To input grade to SIKAD
		get_grade_from_siakad	To get grade from SIKAD
Configure Setting	MOODLE	tree_service	To get list from category and course and create tree
	SIKAD	get_fakultas_siakad	To get data faculty
		vl_store_service	To get data department and study program

Figures 5.7 until 5.10 show functionalities (course enrolment, course registration, student grouping, Learning strategist) with services in the form of sequence of activities.

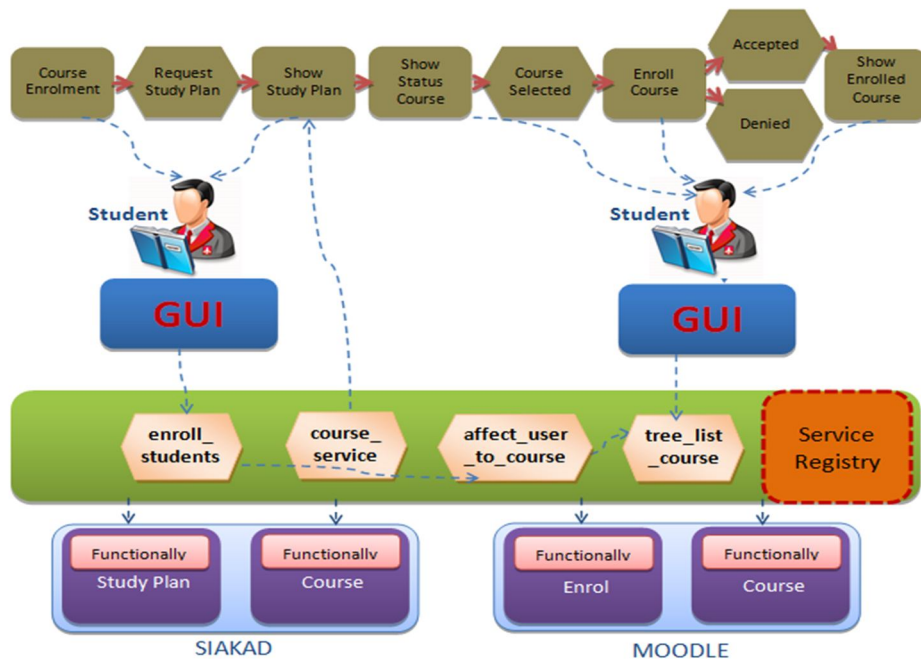


Figure 5.7 Course Enrolment (SIMPEL activities and services)

Students can enroll courses based on their valid courses in study plan. The status of course 'registered or not registered' will be shown in each course in student's study plan. Students can enrol only the courses that have registered status (registered by the teachers).

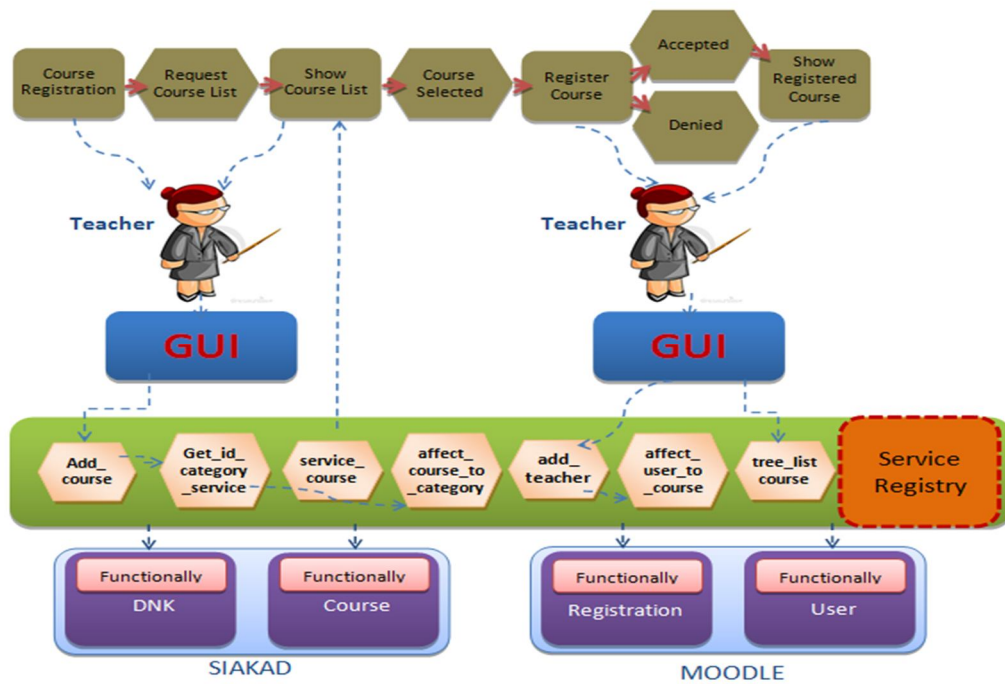


Figure 5.8 Course Registration (SIMPEL activities and services)

The teachers as coordinator courses will see a list of their courses in the on-going semester. Teachers can register courses based on the list of courses. Courses that have already registered will have status 'registered' so that students can enroll the courses.

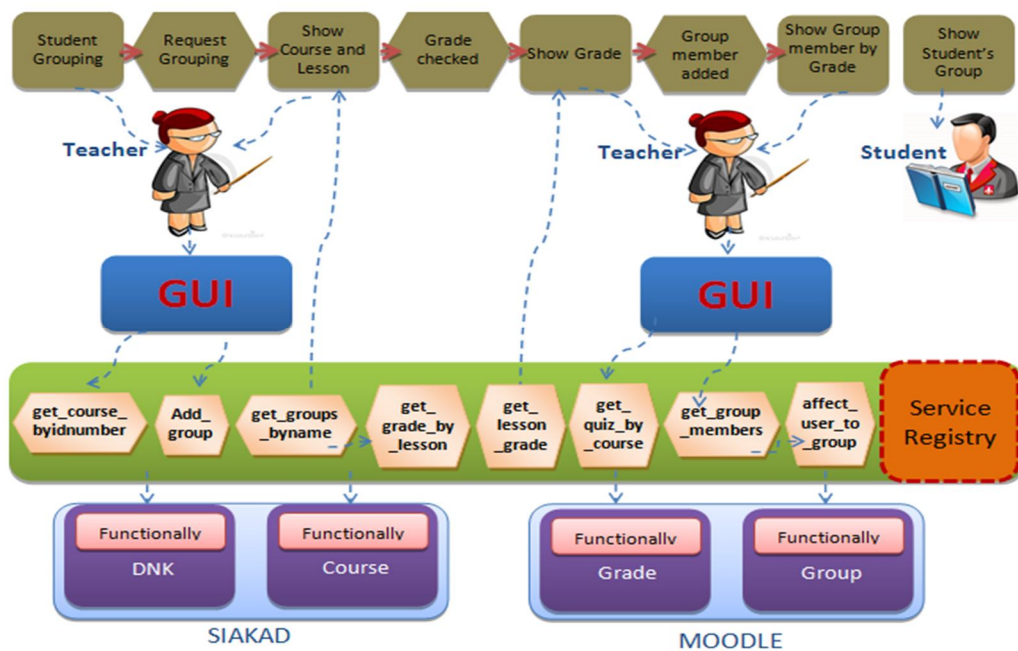


Figure 5.9 Student Grouping (SIMPEL activities and services)

The teachers as coordinator courses will see list of their courses in the on-going semester. Teachers can select courses and group of the students. Student's grouping based on grade automatically done by system.

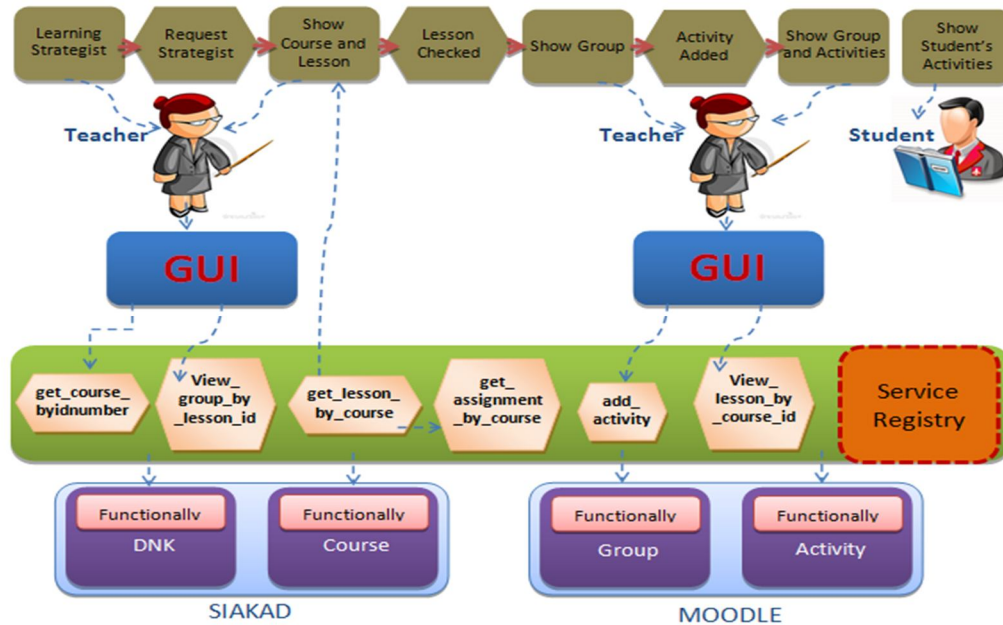


Figure 5.10 Learning Strategist (SIMPEL activities and services)

The teachers as coordinator courses will see a list of their courses in the on-going semester. Teachers can select their courses and advise appropriate learning strategies to students. Learning strategies are determined based on differentiation on group of students.

5.3.4 The generated Web Application

5.3.4.1 Web application based on MVC Pattern

The next step should be done before generating web application is designing the model. As an example of MVC model Figure 5.11 shows class diagram for SIMPEL consisting of classes for CourseDatum, UserDatum, GroupDatum, GradeRecord. For naming class, each class controller added by "Controller", class model added by "Model" and class view added by "View". For example CourseDatumModel, CourseDatumView and CourseDatumController.

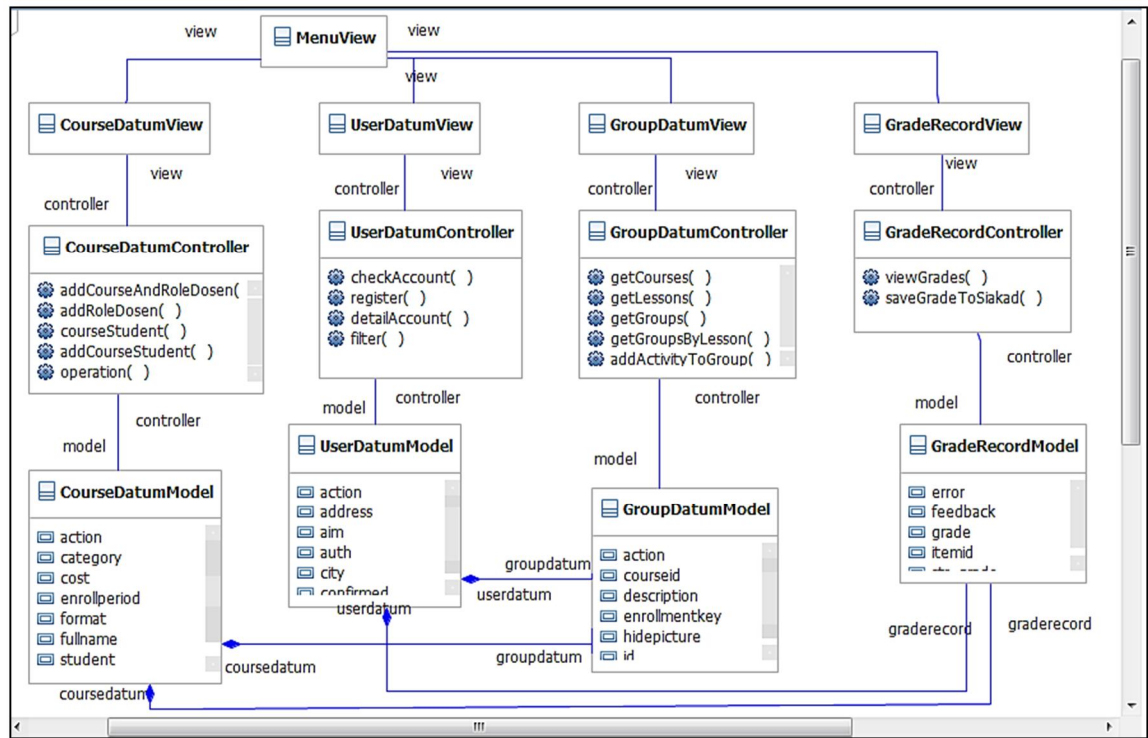


Figure 5.11 An example of MVC web application model

After designing the model, by using templates that have been described in the previous chapter, the next step is generate code. The result of the generated code is PHP files in keeping with the template in use. For example in Figure 5.12., if the model and the template used are Controller so that the resulting files are PHP files for Controller. As can be seen in the figure, the Controller Model and Controller Template used to generate PHP file for the Controller. Number of generated PHP files is equal to the number of classes in the model, because in this case the template defines one class for one PHP file. In this case 8 PHP files are generated based on Controller Model.

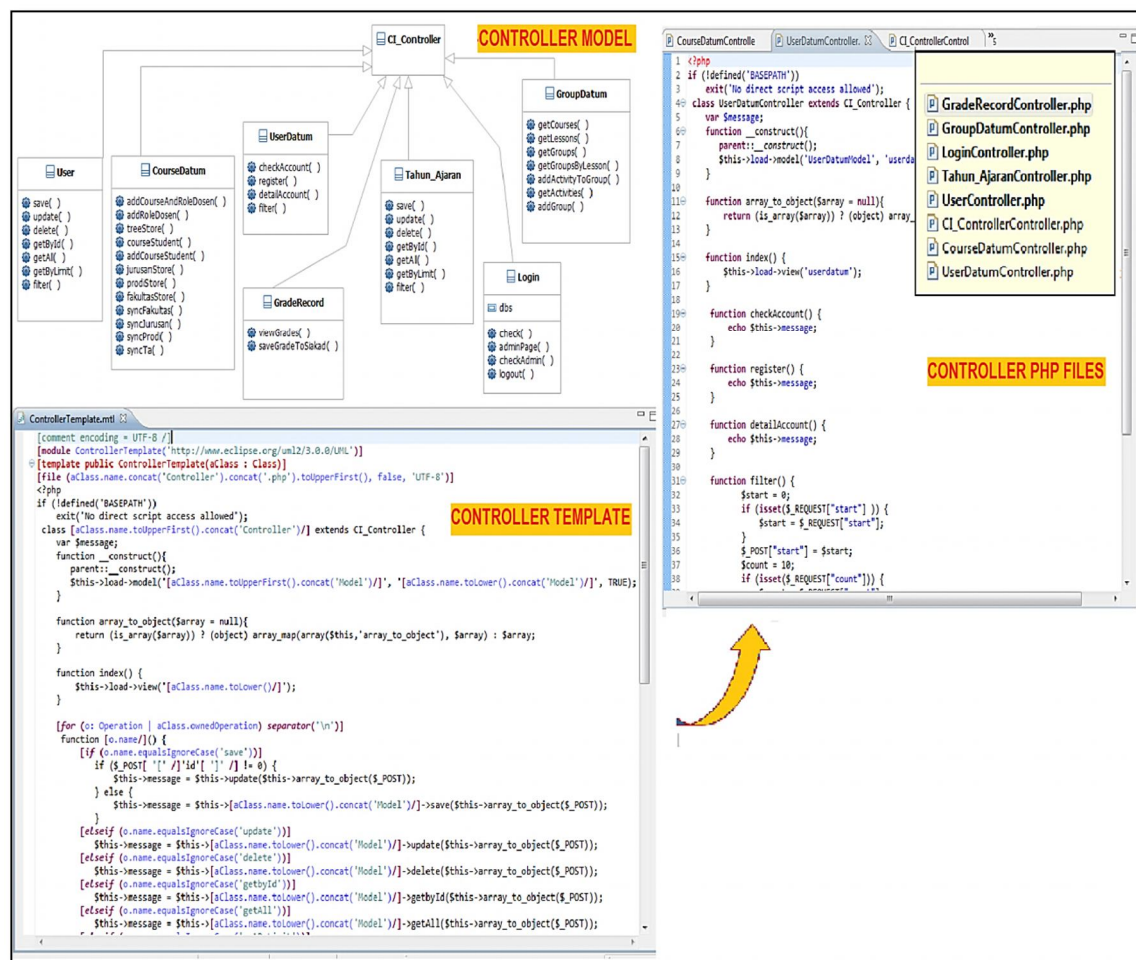


Figure 5.12 Illustration of Model and Controller Template to generate PHP files

All generated files are then completed and customized as needed. The result is the PHP files MVC_Model, MVC_View and MVC_Controller that will be tailored to the structure and folder of web framework CodeIgniter. Implementations of the MVC web applications are displayed in browser. To display the information, the View is rendered by the Controller. The Controller obtains inputs through GET or POST, handing the data based on the Model which comprises specific rules and procedures for handling and processing data. The Controller will use the Model for the processing the data from database and web services, then the new information will be sent to the View, which will then display the information based on templates.

As an example of web page in SIMPEL which use CRUD database operation, a management feature for users (administrators) is illustrated in Figure 5.13. It is applied for managing user's

database. SIMPEL only handles users with an administrator role, while other user roles are managed by SIAKAD. A web page in this figure generated by using MVC CRUDModelTemplate, CRUDViewTemplate, MenusViewTemplate and ControllerTemplate.

Name	Username
Denny	admin
Mardiana	administrator

Figure 5.13 The CRUD UI for updating the database

The figures below also provide some example of SIMPEL web pages : registration and enrolment course. They will describe two features of SIMPEL which illustrate an example of database operation and web service functions. The design of model for this case described in Figure 5.14. This model used as an input for automatic code generation step.

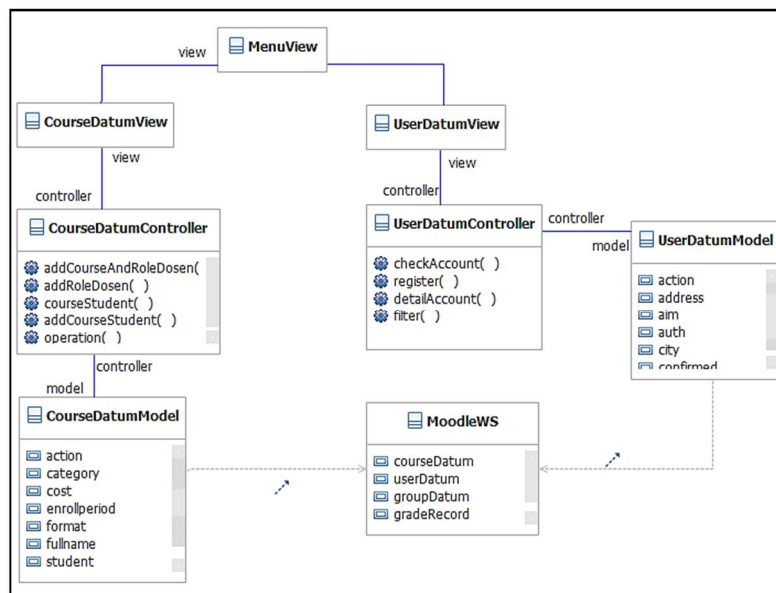


Figure 5.14 Model for the Registration and Enrolment Course

Figure 5.15 shows a different example for web page which use CRUD database operation, a registration page. This web page used to student to register to MOODLE by using their data in SIAKAD. The data of student are displayed in the form registration and student allows to edit and complete some of the data.

Figure 5.15 The CRUD UI for Registration

On the other hand, Figure 5.16 provides an example of a WS UI for course enrolment through SIMPEL. The WS UI generated by using WSMoelTemplate, MenuViewTemplate, TreeListViewTemplates and ControllerTemplate. This web page used to view data obtained from web services of existing application systems. Users can then submit their actions based on available information. As can be seen in Figure 5.16, this figure shows the SIMPLE web page for course enrolment that students can access to register for courses available at MOODLE according to their study plan as arranged in SIAKAD. To the left of the figure, a data group tree is shown consisting of faculty, department, and the academic year. To the right is a list of data that presents courses in the student's study plan, course status and student status, including a button for course enrolment. The list of data related to course enrolment displays data obtained from the SIAKAD web service (get_siakad_course_mhs) and the MOODLE web services (get_course, get_categories, add_course, add_student and affect_user_to_course). As an example of the source code of PHP file for the course enrolment (from MVC_model) is shown in Figure 5.17.

Registration Course Activities Log Out Welcome 0815031040 / IMAN VIVAHIS , Login as MAHASISWA, Academic Year 2011/2012, Even Semester

Course Registered in e-Learning Course Siakad

Course Code	Course Name	Semester	Course Status	Student Status	Action
TEL300	ETIKA PROFESI	6	Available	N/R	Enroll Course
TEL324	ANTARMUKA DAN PERIPHERAL	6	Available	Member	.
TEL320	PENGOLAHAN SINYAL DIGITAL	6	Available	N/R	Enroll Course
TEL342	PEMROGRAMAN BERORIENTASI OBJEK	6	Available	N/R	Enroll Course
TEL346	ORGANISASI DAN STRUKTUR KOMPUTER	6	N/A	N/R	.*
TEL104	PEMROGRAMAN KOMPUTER	2	Available	N/R	Enroll Course
TEL350	STRUKTUR DATA	6	N/A	N/R	.*

Notice :
 .* = Course Not Available
 N/A = Not Available
 N/R = Not Registered

Figure 5.16 The WS UI for Course Enrolment

```
function setObject($obj = null){
    $courseDatum = new courseDatum();
    $courseDatum->setTeachers($obj->teachers);
    $courseDatum->setStudents($obj->students);
    $courseDatum->setTeacher($obj->teacher);
    $courseDatum->setCategory($obj->category);
    $courseDatum->setFormat($obj->format);
    $courseDatum->setPassword($obj->password);
    $courseDatum->setIdnumber($obj->idnumber);
    $courseDatum->setMaxbytes($obj->maxbytes);
    $courseDatum->setFullname($obj->fullname);
    $courseDatum->setStudent($obj->student);
    $courseDatum->setShortname($obj->shortname);
    $courseDatum->setShowgrades($obj->showgrades);
    $courseDatum->setSummary($obj->summary);
}

function get_course($obj = null) {
    $res=$this->client->get_course($this->lr->getClient(),$
    return $res->getCourses(); //getCoursesReturn
}

function get_teachers($obj = null) {
    $res=$this->client->get_teachers($this->lr->getClient()
    return $res->getUsers(); //getUsersReturn
}

function get_categories($obj = null) {
    $res=$this->client->get_categories($this->lr->getClient
    return $res->getCategories(); //getCategoriesReturn
}

function add_course($obj = null) {
    $course = $this->setObject($obj);
    $res=$this->client->add_course($this->lr->getClient(),$t
```

Figure 5.17 An excerpt source code for the enrolling course (CourseDatumModel.php)

5.3.4.2 DDL (Data Definition Language)

All the generated PHP files in the SIMPEL application are in form Model, View, Controller. These PHP files produce a set of web pages for the integration of the SIAKAD and MOODLE. However, the SQL files are also generated by using DDLTemplate for creating the required

database. As an illustration of DDLmodel and DDLtemplate to SQL file can be seen in Figure 5.18.

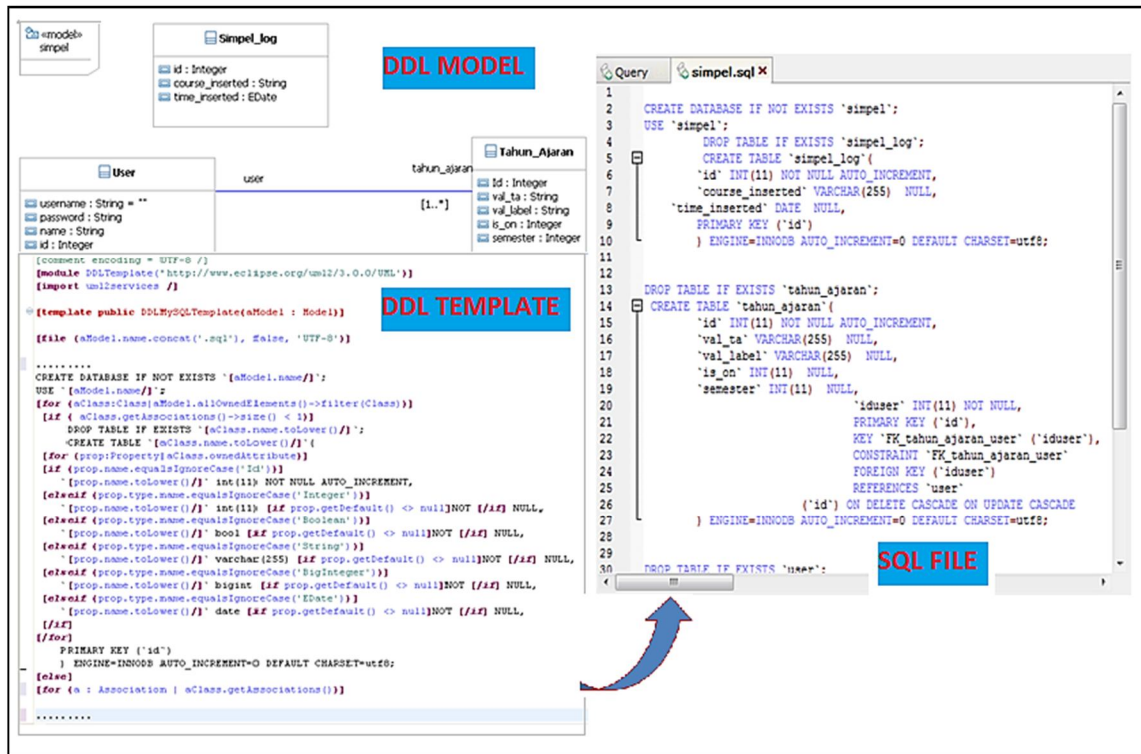


Figure 5.18 Illustration of DDL model, DDLTemplate and SQL File

Figure 5.19 illustrates the generated all files for the SIMPEL application. The structure of files is divided in three main folders : Application, Database and WSDL. The last one contains the WSDL files and PHP files to define the interface to web services. All the generated WSDL dan PHP files in this folder are then configured by the related server. The folder Database contains the DDL SQL script that allow us to create the data base required for the web application. All the generated DDL files in this folder are then executed by database server. The folder Application contains the folder of the Models, Controllers and Views. For the implementation all the folders and files are tailored to the web framework Codeigniter.

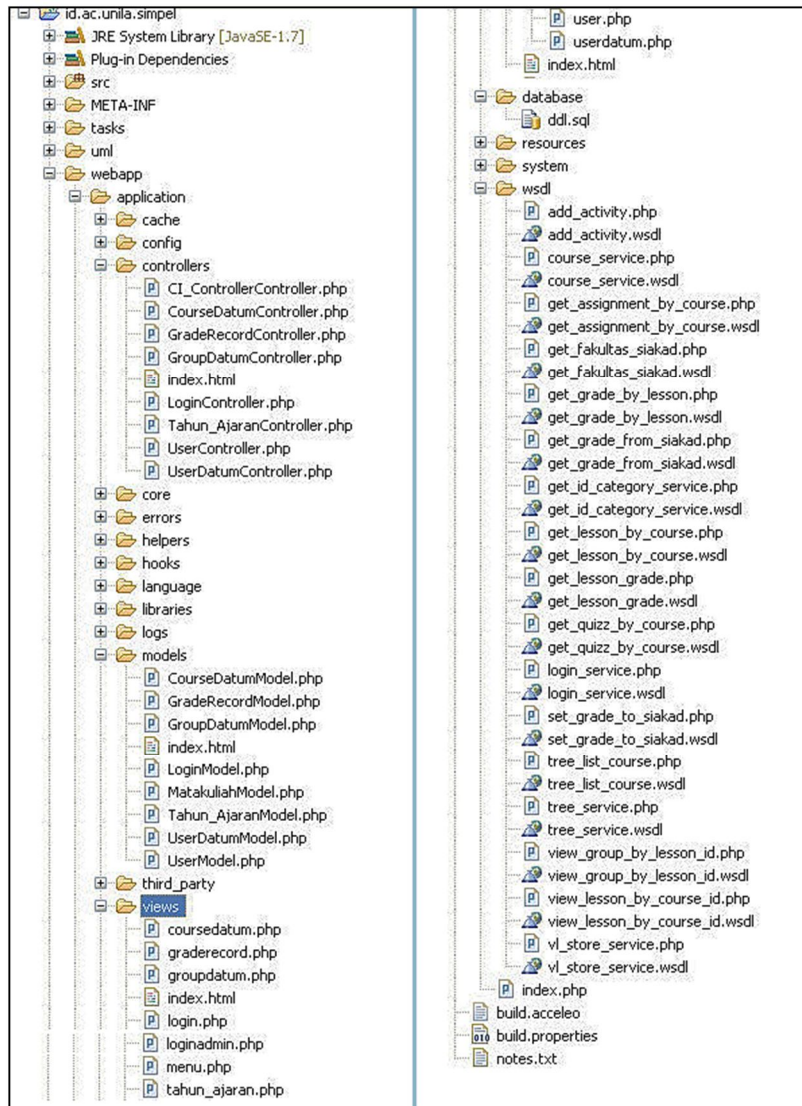


Figure 5.19 Generated files for the SIMPEL application

5.4 The SIMPEL Implementation

5.4.1 Demonstration Scenario

Furthermore a more detailed scenario of interaction students and teachers with the system will be described in the following section. To show the sequence of activities all the examples of pages are noted by numbers.

The following section provides a demonstration scenario for student and teacher users. Necessary data on student-related activities must first be create available in the existing applications, as provided below:

SIAKAD: At the start of each semester, students create their own study plan in SIAKAD through consultation and validation from their respective academic advisor.

MOODLE: Each teacher prepares their respective courses with regard to learning material, tasks, exercises, and online tests for students via MOODLE. The respective course topic will display all different types of available e-learning activities.

5.4.1.1 Student Side

Users login to SIMPEL with their own SIAKAD account. Only users authenticated by SIAKAD can access SIMPEL. As seen in the main menu presented in Figure 5.20, the web page allows students to register the MOODLE, enrolment, select activities, view group and grade. The illustration of each menu is as follows:

(Point 1). When a student registers to MOODLE, the system will do a check on whether the student is already registered or not. If the student is not found in the MOODLE's database, the registration page will appear, prompting the student to fill out the displayed form. The data profile of student are obtained from SIAKAD and displayed in the form. Student allows to edit and complete data appear except the username, first name and surname.

(Point 2). Students create their study plan in SIAKAD at the beginning of semester, consult and get validation of their plan from their advisor. After the validation process, all the courses in study plan will be shown for the enrolment process in SIMPEL. Students may select each course available to enroll to e-learning activities. The courses that can be enrolled by students must be registered by the teacher on the current semester.

(Point 3). The enrolled course is then selected in order to enable students to perform activities provided through the MOODLE application, such as reading of course material, dialogue, group discussion, project work, assessment, and course evaluation.

For each lesson, students must take an online test after which they will be informed of their learning competency. Students are later grouped according to earned grade range.

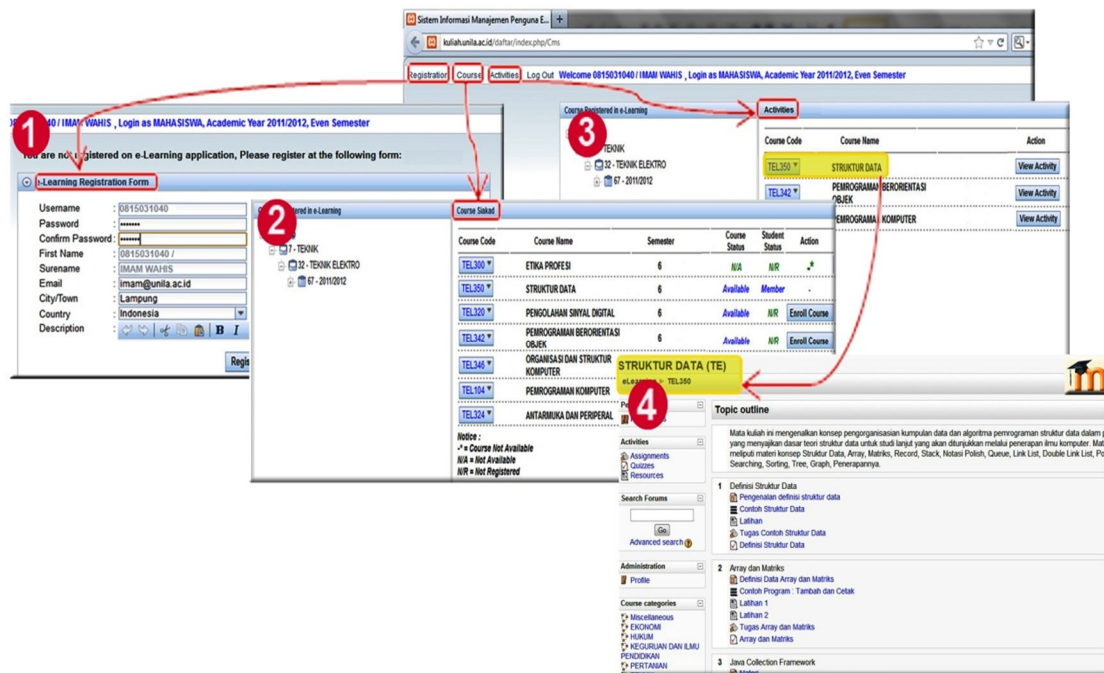


Figure 5.20 Demonstration of the interaction student

5.4.1.2 Teacher Side

In the beginning of semester, the information of schedules including the responsible teachers and assistant are stored in SIAKAD. The courses and its teachers are shown in SIMPEL for registration process. Then the registered course can be selected in order to prepare learning material, task, exercise, and online test for students in MOODLE. The student takes online test at the end of each lesson and their grade will be used for grouping the students.

(Point 1). Basically, MOODLE can facilitate the teacher to create group of students manually, but there is also an extension to create the group of students automatically based on student's grade in each online test.

(Point 2). Based on that grouping, teacher will have information about mastery level of learning material. This information help teacher analyze the effectiveness of that learning

material and adjust learning material content, task, exercise, support, and time allocation adjustment given for each group of student.

(Point 3). At the end of semester, teacher can calculate total grade from online test and offline activities.

(Point 4). Then the final grade of the course is stored in SIAKAD directly. Therefore teachers do not have to input grade manually like previous process. Teacher interaction toward system is shown in Figure 5.21.

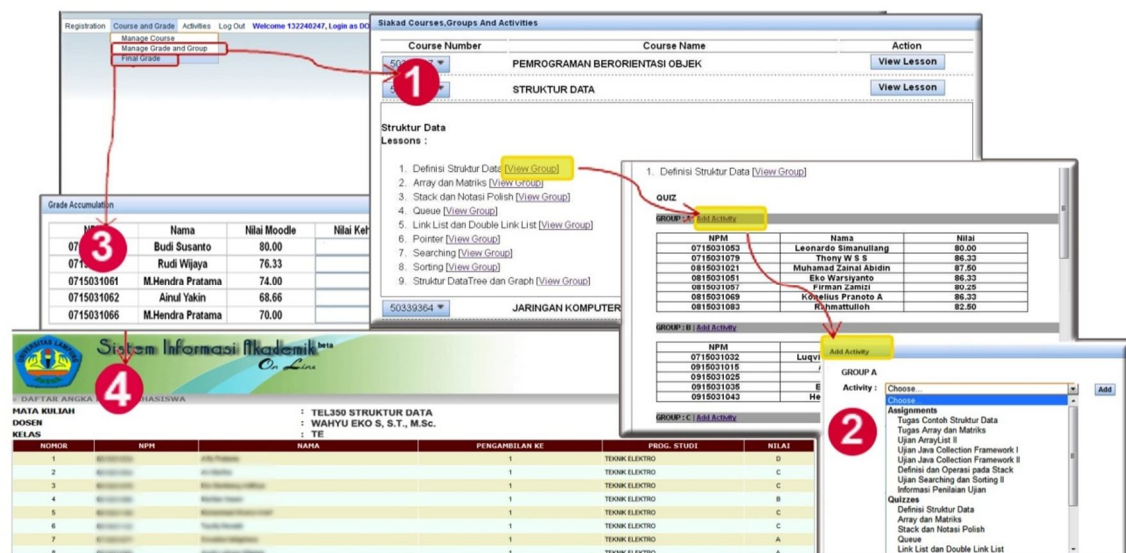


Figure 5.21 Demonstration of the interaction teacher

5.4.1.3 Other Feature

Configuration of basic setting for preparation at the beginning of each semester, such as year, semester, start and end date, is done by the administrator. One benefit from the integration of information is a combination of data from two different systems can provide new information. SIMPEL in this case also presents statistic and report for administration purposes i.e. for showing number of registered courses, course enrolment, active courses and students presentation in each week containing learning material.

5.4.2 Evaluation of SIMPEL

5.4.2.1 Evaluation of SIMPEL implementation

The implemented SIMPEL is analyzed in the teaching/learning activities at Department of Electrical Engineering Unila on trial period in odd and even semester of the 2011/2012 academic year. During this period a total of 887 students were enrolled in 25 courses through the SIMPEL and follow the learning activities in MOODLE. As the result the teaching/learning activities can be displayed properly by the synchronization of the MOODLE and the SIAKAD. Users have gained great benefit from the integration of an academic information system and CMS. The advantages of this integration can be illustrated by comparing the before and after situation of the SIMPEL. Before the launch of SIMPEL the users need multiple authorizations to access available systems (SIAKAD and MOODLE), requiring them to separately create an account and enroll in courses for each system.

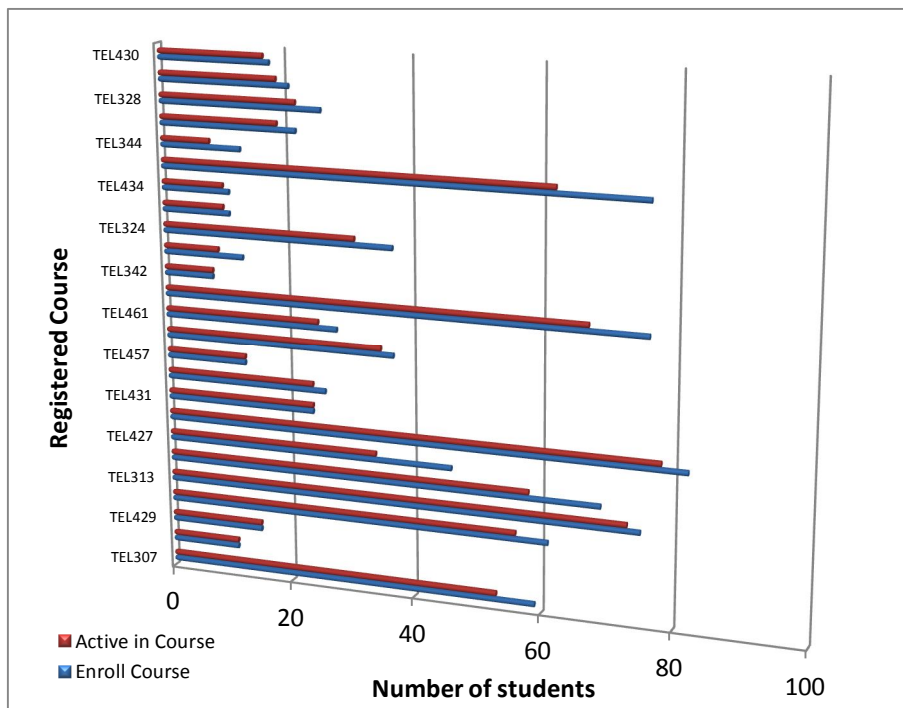


Figure 5.22 Number of enroll course and active student in course

Furthermore, there is lack of data synchronization for each academic semester, and teaching staff must manually create groups and work out grades. With the SIMPEL the users have single

authorization where they can create an account and automatically enroll for courses based on data available in other systems. As a consequence, data is synchronized for each semester and teaching staff can automatically create groups based on grade range. These complementary features have been implemented to enrich the academic activity.

Figure 5.22 shows number of students doing course enrolment and shows how many of them active in doing activities at each registered course. As seen in this figure, number of active students in the course is less than the enrolled number. The trend is occur in almost all of the courses, despite those discrepancies are not very significant. Moreover, it can also be seen that the discrepancy is not occur at the five courses with the small number of enrolled student.

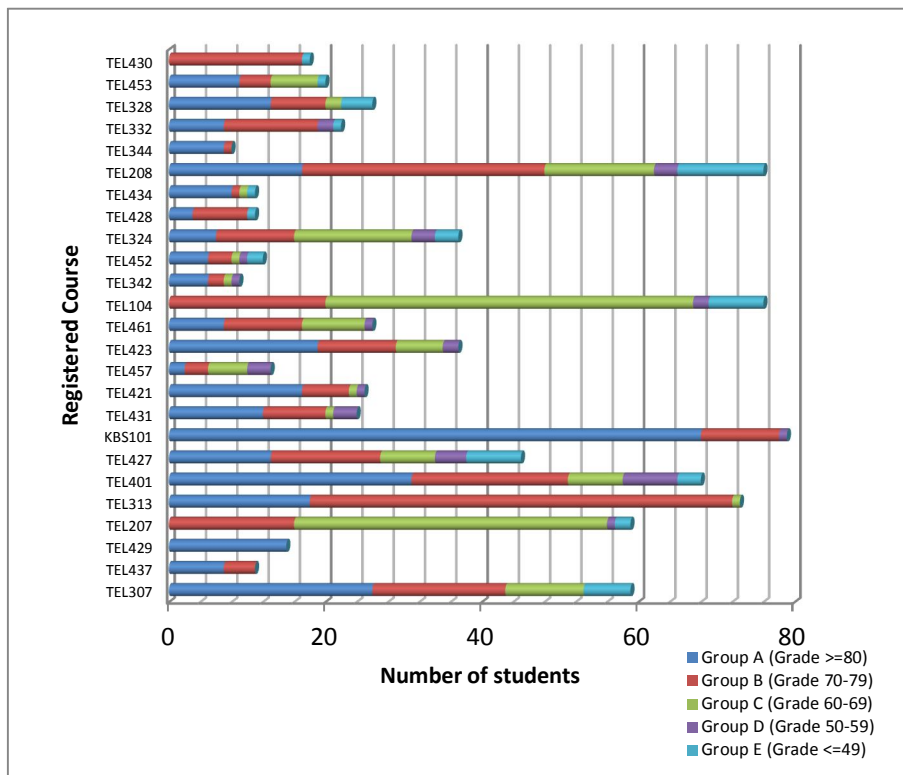


Figure 5.23 Distribution of students in each group

Additionally, the distribution of students in each group is shown in Figure 5.23. They are grouped by range of grade obtained in the first online test of the lesson each course. The groups are divided into five categories A,B,C,D,E based on the range of their score. Group A is for students with the highest grade and group E for the lowest. The figure depicts that

students are grouped into group D and E with a range of score 0 to 60 still have smaller percentage than the other groups. This means students have a good ability to understand their first material. Moreover in the course code TEL429 all of students are grouped into A categories. Evaluation process at SIMPEL is useful for teacher to help understand student's mastery level of learning material and therefore it can help teacher analyze relevance and effectiveness of given learning material. That information then will be used to decide whether the learning material will be kept or replaced by more appropriate material based on student performance assessment. The result showed that SIMPEL can properly perform its functionalities to guide students and teachers in their activities.

5.5 Chapter Summary

In this chapter we have presented a development of web information system by applying the MDA-SOA approach to integrate two existing information systems. A case study conducted by designing the SIMPEL as an web interface between academic information systems and CMS at Unila. This approach allows developers go from information system model at the beginning to generate web application automatically. The developer also can define their own functionalities which not cover yet in the previous systems. It can also assist in the development of web application that invokes web services to enable the interoperability of two existing web application. The results have shown that this approach can be applied and work properly.

The implemented SIMPEL is analyzed in the teaching/learning activities at Department of Electrical Engineering Unila on trial period. The evaluation result showed that SIMPEL can properly perform its functionalities to guide students and teachers in their activities. This work provided a concrete benefit for bridging the two important web information systems at Unila as well as decreased budget allocation from commercial integration solution. From the developer viewpoint, the process and the technologies adopted is expected can be reused to develop a different design and the target platform.

Chapter 6 Evaluation Model for WIS Quality and Effectiveness

This chapter presents an evaluation model for evaluating WIS that allows university to assess quality and effectiveness of developed and implemented WIS. This model is constructed by modifying the concepts of both Technology Acceptance Model (TAM) and IS-Impact due to inadequate comprehensive evaluation for WIS provided by traditional IS evaluation. The evaluation model then applied to assess the quality and effectiveness of WIS development and implementation in Unila. It is described in two parameters: performance evaluation and functionality evaluation.

6.1 Chapter Introduction

Evaluation is an important aspect necessary for gleaning information on the extent to which the objectives of a WIS has been successfully achieved. Performance of an HEI can be enhanced by the WIS. However, all the advantages of the WIS become useless if the WIS is not used properly by the user. Therefore, it is significant for developers to understand how the acceptance of users toward the developed WIS. Whether users accept the system and decide to use or not use the system. Through evaluation, feedback from users will also be generated, crucial for improving WIS quality in the future.

The main constraint in conducting evaluation is in determining the evaluation criteria, evaluation parameter and methodology for establishing the evaluation framework. Recent

developments in IT, particularly with regard to the internet, have made it even more difficult to gauge the level of success and effectiveness of an WIS compared to earlier conditions. WIS has brought forth a different set of conditions and factors that need to be considered, such as direct user interaction unrestricted by time, distance and place. As already described in the previous chapter, the purpose of the organization is an important construction that should be considered when evaluating WIS. Because the measure of success of WIS should add value to the organization's goals. The existence of internet technology has expanded the context and content of assessment of an information system. WIS however has changed the organization's internal business processes, such as giving more time savings and efficiency. Based on these effects, in this chapter, a model for evaluation of WIS quality and effectiveness assessment will be explored.

In this study, we seek for the possibility of constructing a new model for evaluating web information systems by applying the concepts of both Technology Acceptance Model (TAM) and IS-Impact. The proposed model adopts the assessment of the Critical Success Factors (CSF) affecting WISs in HEIs to demonstrate how the proposed model can be applied and supporting the decision makers in developing and evaluating the WISs.

6.2 WIS Evaluation Model

WIS evaluation model intended for measuring of WIS quality and effectiveness. It is also a measure of the success of the WIS. Success measure in this context is how the impact of WIS related to the existence of the environment. The environment can be internal and external environment of HEI. For internal environment of HEI, WIS is directly related to the users, the existing systems and working units within the HEI. As for the external environment, WIS associated with the users and other systems. Linkages with both internal and external users aim to facilitate the users in understanding and accessing the system. Interactions of WIS with existing systems in both web-based system and non web-based system to be an important aspect of assessment. WIS should have a good impact, support and even improve the system performance.

Every HEI has its own specific strategies identified by management. These strategies set out in the CSF institutions. The CSF is an area that indicates the success of the performance of a

working unit within the HEI. Every HEI has different CSF depending on what elements can determine success or failure in achieving the goals of the institutions. In this context, the success of WIS assessed on all aspects of the WIS satisfaction in contributing to achieve CSF in HEI. Satisfaction of users and other stakeholders is important to note because they are an important element for the success of an HEI.

6.2.1 A Reference WIS Evaluation Model

The WIS literature provides several definitions and measures of WIS quality and effectiveness. From previous studies, several recommended models for information system evaluation include Technology Acceptance Model (TAM) and IS-Impact. TAM introduces two key variables - Perceived Ease of Use (PEU) and Perceived Usefulness (PU) – that have central relevance to predict user acceptance [82][83]. Through the IS-Impact model, IS impact can be measured in terms of Information Quality (IQ), System Quality (SQ), Individual Impact (II) and Organizational Impact (OI) [84]. Figure 6.1 shows TAM and IS-Impact model.

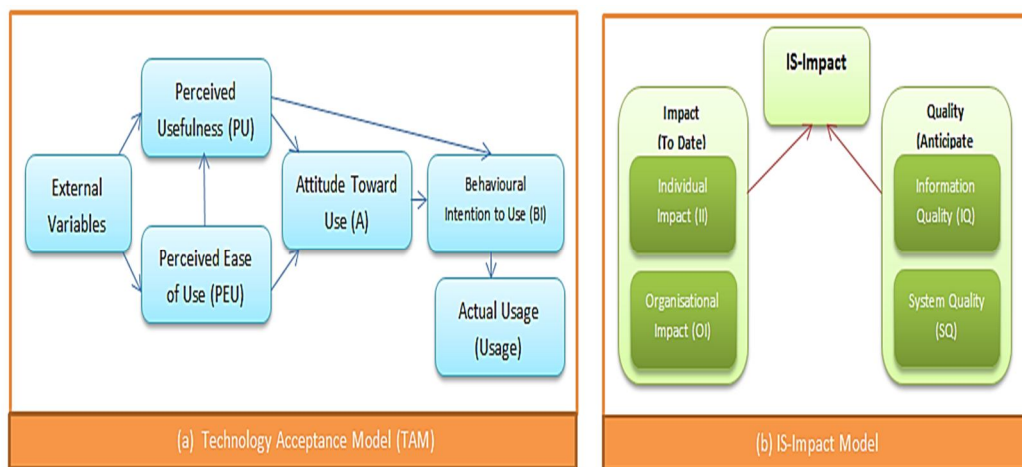


Figure 6.1. (a) TAM and (b) IS-Impact model

In Figure 6.1 (a) TAM states that PU and PEU, are relevant factors and have a significant impact on computer acceptance behaviours. PU can also be defined: $PU = PEU + \text{External Variables}$

External variables may affect usefulness directly or indirectly through perceived ease of use. In cases where similar systems are equally easy to operate, external variables may be the critical

determinant of perceived usefulness. External variables refers to system characteristics, user interface characteristics, user training, and user computer experience. PEU can influence attitudes and behaviour in two ways: self-efficacy and instrumentality.

In term of web-based system, author in [85] presented TAM model in order to predict the use of Web-based information systems by extending the model with some variables (elements). However the model has shortcomings that have emerged in the context of previous research such as: this model has to be integrated into a broader model [86]; this model has only focus on the acceptance by students in educational institutions [87][88] but acceptance within organizations is rarely covered [89].

Generally, although those models are widely used for explaining the acceptance of information systems, both have some limitations. In view of this, the evaluation model for web information system implementation is modified from the aforementioned models by incorporating specific measurements. The selection of the two existing models and the necessary modifications are essential with regard to WIS for the HEIs in which the characteristics are different from other enterprises.

Consequently, 12 dimensions is proposed for measuring WIS quality and effectiveness : Individual Impact; Organizational Impact; Information Quality; System Quality; User Capabilities; Organisation Capabilities; Information Effectiveness; System Effectiveness; Perceived Usefulness; Perceived Ease of Use; Behavior Intention; User Satisfaction. The model assumes that Individual Impact, Organizational Impact, Information Quality and System Quality are linked to User Capabilities; Organisation Capabilities; Information Effectiveness; System Effectiveness, and these in turn; influence Perceived Usefulness and Perceived Ease of Use which affect on Behavior Intention and User Satisfaction as shown in Figure 6.2.

Users will accept or reject the system based on their belief that the system can help them perform their tasks better. Perceived Usefulness and Perceived Ease of Use are user's subjective assessments of performance and effort, do not automatically reflect objective reality. In case users do not perceive the system as useful or easy to use they will possibly not use it.

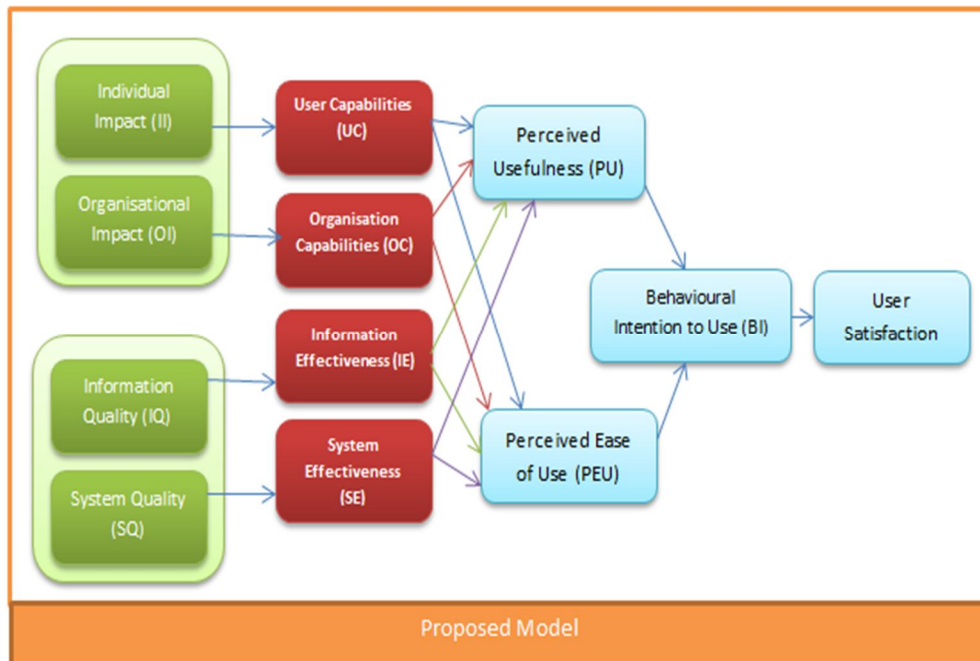


Figure 6.2. Proposed evaluation model of WIS

6.2.2 Measurement Dimensions

The defined dimensions for measuring WIS by adding 4 dimensions : User Capabilities (UC), Organisation Capabilities (OC), Information Effectiveness (IE) and System Effectiveness (SE). Table 6.1 show the dimension and elements for measuring the dimensions.

Table 6.1 Dimensions and elements for measuring the WIS

Dimension	Element
Individual Impact (II)	Enjoyment , Training, Functions
Organizational Impact (OI)	Cost reduction, Time savings
Information Quality (IQ)	Functions , Timelines , System features, Improving work efficiency, Interactivity, Improved decision making
System Quality (SQ)	Availability, Maintainability, Reliability, Server and Network Speed, Assurance, Security, Completeness, Relevance
User Capabilities (UC)	Ease of use, Ease of learning
Organisation Capabilities (OC)	Enhancement of communication, Enhancement of coordination

Information Effectiveness (IE)	Navigation patterns, Conciseness, Services, Consistency, Responsiveness, Applicability, Integration Interactivity, Number of academic standards procedure supported, Number of operational university supported
System Effectiveness (SE)	Accessibility, Traceability, Response time, Currency
Perceived usefulness (PU)	Responsiveness, Security
Perceived ease of use (PEU)	Accessibility, Ease of use, Response time, Ease of learning
Behavior Intention (BI)	Number of site visits, Response time, Number of transactions executed, Empathy, Integration
User Satisfaction (US)	Enjoyment, Empathy

Based on the characteristics of the HEI, the dimensions used still needs to be adapted for specific user. For example, the OI dimension is not provided to the user student due to not directly related to the organization impact. The elements for measuring the dimensions consist of 19 elements for students, teachers and related working units user, 10 elements for System Developer, 10 elements for Managements as shown in Table 6.2.

Table 6.2 User and related element

User	Element
Students, Teachers, Staff related working units	Enjoyment, Number of site visits, Number of transactions executed, Availability, Reliability, Accessibility, Response time, Ease of use, Ease of learning, Navigation patterns, Functions, Training, Services, Empathy, Responsiveness, Assurance, Security, Integration, System features
System Developer	Conciseness, Timelines, Consistency, Maintainability, Applicability, Server and Network Speed, Traceability, Security, Currency, Interactivity
Management	Cost reduction, Time savings, Improving work efficiency, Enhancement of communication, Enhancement of coordination, Improved decision making, Completeness, Relevance, Number of academic standards procedure supported, Number of operational university supported

6.3 Measurement Results

6.3.1 Performance Evaluation

This section gives the performance evaluation of SIMPEL development and implementation. For the evaluation of SIMPEL development we evaluate the result by evaluating the models, the defined templates, as well as the handled generated code and handwritten code. We

measured the LOC (Lines of Code) of generated files and also finished files that already fix after adding handwritten code. In addition, we evaluate the SIMPEL implementation in trial period in Unila.

6.3.1.1 Performance SIMPEL development

Measurement results are provided in Table 6.3. Measurements are done on LOC of files for each Model, View and Controller. LOC refers to the number of the finished code after adding handwritten code while GLOC is the number of generated lines of code. HLOC refers to the number of handwritten of code while %HLOC is the percentage of LOC that is handwritten. Then %THLOC column presents a percentage of LOC of the total number of LOC that is handwritten. As a standard formula for line counters in web pages, every line of scripts or expressions will count as one line and XML defined or self-defined tags will similarly count as one line. In this case, we only consider PHP files that generated by using the defined MVC templates not include the other supporting files or framework. As presented in Table 6.3, most handwritten lines of code are contained in the MVC_view is about 30.2 %. This substantial amount handwritten line of code in MVC_view due to the addition of code such as image, table and PHP logic which may suggests that MVC_view is part of the application where the best optimization level can be achieved.

Table 6.3 Measurement result of SIMPEL

PHP FILES	LOC	GLOC	HLOC	% HLOC	% THLOC
MODEL	615	490	125	20.3%	3.8%
VIEW	2073	1446	627	30.2%	19%
CONTROLLER	618	444	174	28.2%	5.3%
TOTAL	3306	2380	926		28.1%

Productivity can be estimated from the amount generated line of code and handwritten lines of code. It can be deduced that the more lines of code generated, the higher the increase in productivity, due to less effort in manual coding. The total handwritten code is about 28.1 % of the total application was generated (%THLOC). A productivity increase is expected once the amount of handwritten lines of code is reduced. Similarly, productivity also related with the time effort for the development. In this case, development productivity is measured in FP (Function Point) per hour. We compared the time required by conventional development

SIMPEL before using the MDA and after redevelop with MDA. The SIMPEL development in the previous work was done conventionally (through writing the code manual). Table 6.4 shows the time effort to realize FP for conventional development and MDA development. As provided in Table 6.4, time effort for the conventional development took a total of about 320 man-hours, realizing 107 FPs and resulting in a productivity of one FP per 2.99 hours. Then the SIMPEL application was redevelop using MDA and handwritten coding in 128 man-hours. Available data shows a productivity increase of 2.5 times faster compared to conventional development.

Table 6.4 Time development of SIMPEL

Development	FP	Time (H)	H/FP	FP/H
Conventional SIMPEL	107	320	2.99	0.33
MDA SIMPEL	107	128	1.19	0.83

Base on the results from the case study it is discovered that the models and the defined templates have potential for developing a web application interface with a high productivity.

Results from the case study have definitely illustrated this approach potential for developing web applications with high performance. However, the performance increase can be gained by extending the templates and models. The first area for improvement is template extension to deal with the complex and repetitive tasks involved in web application development, such as database operation and framework configuration. Then models can be extended further for the purpose of achieving a greater amount of code generation. This will result in extra code generation due to greater model details, but may also prompt redundancy issues. Another improvement to the method relates to the flexible managing of both generated and handwritten codes, whereby changes to the templates lead to changes in the generated code. A problem may however arise which requires the developer to change the handwritten code due to template changes. This can be avoided by separating handwritten and generated codes which will also decrease the coupling of both generated and handwritten codes. Furthermore, once these flexibility issues are resolved, a higher overall productivity level is expected.

6.3.1.2 Performance SIMPEL Implementation

The proposed evaluation model is intended to address key measures of implementation. As can be seen in Table 6.5 the SIMPEL quality measures has presented. The rated was given by the developer and administrator. Result column contents are filled based on the operational observation. It is shown in the table that adding or improving WIS functions, although practicable, is still not an easy task for the developer. Percentage of EPSBED data and QA main data unsupported are 30%. This is due to unrelated functions between e-learning system and EPSBED data or QA. In line with this, additional database tables, web services, user interface and functions are necessary in order to improve performance.

Table 6.5 SIMPEL quality measures

Measure	Result	Maintenance Procedure
Unsupported academic procedures	zero	Adding web services & functions.
Unsupported e-learning procedures	zero	Adding web services & functions.
Percentage of EPSBED data unsupported	30% ^(a)	Adding database tables, web services, user interface & functions.
Percentage of QA main data unsupported	30% ^(b)	Adding database tables, web services, user interface & functions.
Time requested for system delivery	2 semester	Gradual system functions delivery.
Unsupported reports	1 report /month (users keep requesting more reports) ^(c)	Improving functions & reports.
Percentage ease of adding or improving functions	70% ^(d)	Improving and adding functions

(a) Rated by the developer and administrator.

(b) Rated by the developer and administrator.

(c) Computed using the data obtained from the helpdesk.

(d) Rated by the developer and administrator.

6.3.2 Functionality Evaluation

The model adopted for WIS evaluation is a modification of the IS-Impact and TAM models with 12 dimensions of measurements. This model is applied to assess the quality and effectiveness of SIMPEL implementation in Unila. The designed questionnaire consists of 19 elements for user students and teachers. Given the time constraint in evaluating the implementation of SIMPEL, the evaluation process only involves participants representing user students and teachers and has yet to involve all stakeholders in Unila. Some 358 users have expressed their views through the questionnaire on SIMPEL implementation for the recommended dimensions.

Results reveal that 64.7% of users are satisfied with the SIMPEL (highly agree and agree), neutral 30.4% and 4.9% unsatisfied (disagree and highly disagree) as shown in Table 6.6.

Table 6.6 Functionality Evaluation result

Element	Highly agree	Agree	Neutral	Disagree	Highly disagree
Enjoyment	32.7%	48.3%	16.8%	2.0%	0.3%
Number of site visits	25.7%	55.9%	16.2%	1.7%	0.6%
Number of transactions executed	18.2%	40.5%	37.4%	3.6%	0.3%
Availability	18.3%	50.1%	25.6%	5.0%	1.0%
Reliability	17.6%	43.3%	29.9%	7.5%	1.7%
Accessibility	13.7%	39.7%	37.7%	7.5%	1.4%
Response time	14.8%	42.7%	34.6%	7.0%	0.8%
Ease of use	17.9%	49.7%	27.1%	4.5%	0.8%
Ease of learning	17.0%	51.7%	28.5%	2.2%	0.6%
Navigation patterns	15.6%	51.1%	29.1%	3.9%	0.3%
Functions	17.2%	47.9%	31.3%	3.4%	0.3%
Training	13.3%	47.8%	32.5%	5.4%	1.0%
Services	13.7%	43.9%	37.4%	4.5%	0.6%
Empathy	15.1%	43.3%	36.0%	5.0%	0.6%
Responsiveness	14.8%	39.1%	38.5%	6.1%	1.4%
Assurance	21.6%	42.2%	32.1%	3.5%	0.6%
Security	18.6%	51.4%	27.4%	2.2%	0.4%
Integration	16.8%	52.4%	28.4%	2.1%	0.4%
System features	17.3%	48.3%	31.8%	2.2%	0.3%

Figure 6.3 shows the elements importance perceived by users. Questionnaire results show that three significant elements in SIMPEL are Number of site visits, Enjoyment and Integration. It

show that SIMPEL effectively assist the teaching learning process. While three elements that must be improved are Response Time, Accessibility and Reliability. It shows that SIMPEL still need to be complemented and enhanced the quality.

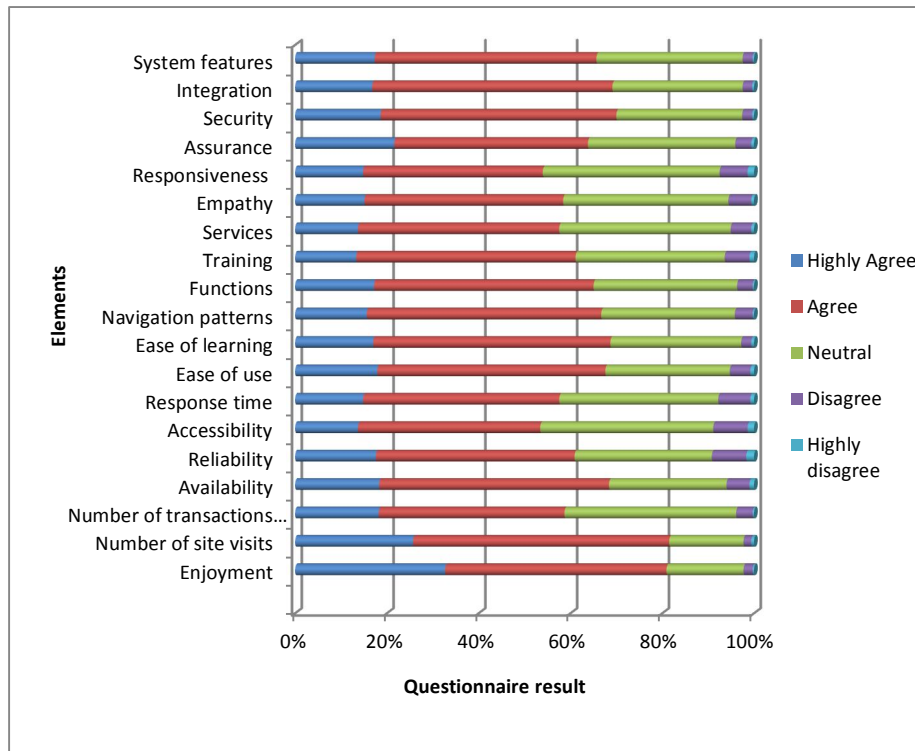


Figure 6.3. Elements Importance

6.4 Chapter Summary

This WIS evaluation model constructed due to inadequate comprehensive evaluation for WIS provided by traditional information system evaluation. In view of this, the WIS evaluation model is modified from the aforementioned models by incorporating specific measurements. Twelve dimensions are proposed for measuring information system. The elements for measuring the dimensions consist of 19 elements for students, teachers and related working units user, 10 elements for System Developer, 10 elements for Managements This evaluation model then applied to assess the quality and effectiveness of WIS development and implementation described in two parameters: performance evaluation and functionality evaluation. The performance evaluation used to evaluate performance of the WIS

development and implementation. In this case the performance of SIMPLE development and implementation.

The performance evaluation result of the SIMPLE development described as follows : Usefulness of the templates to automatically generate PHP files in the form of MVC files, WSDL files and DDL files is powerful enough to cover the constraints of time effort and write repetitive code. However, it would be unfeasible to generate a complete and running web application directly at once. Handwritten codes need to be added for complex business logic. Nevertheless, in this case study the amount of total handwritten line of code 28.1% of total line of code. Time effort to develop application through the MDA is 2.5 times faster than conventional development without MDA. It also results in an increase in development productivity. The performance evaluation result of the SIMPLE implementation described as follows : Unsupported academic and e-learning procedures are zero. It means SIMPEL can cover related academic and e-learning procedures that university need as already defined in requirements specification phase. However percentage of EPSBED data and QA main data unsupported are 30%. In line with this, additional database tables, web services, user interface and functions are necessary in order to improve the performance.

On the other hand, the functionality evaluation used to evaluate the functionality of the WIS implementation. In this case the SIMPLE implementation. The functionality evaluation has been implemented by user based on a questionnaire. Some 358 users have expressed their views. The results indicated that 64.7% of users are satisfied with the SIMPEL (highly agree and agree), neutral 30.4% and 4.9% unsatisfied. It shows that SIMPEL effectively assist the teaching learning process derived from the elements of assessment : Number of site visits, Enjoyment and Integration. However the result also shows that SIMPEL still need to be complemented and enhanced the quality derived from elements of assessment : Response Time, Accessibility and Reliability. The overall evaluation results show that the evaluation model can be valuable for decision makers in HEIs on evaluating the development and implementation of web information systems.

Chapter 7 Conclusion and Future Work

In this last chapter, we present the conclusion and future work of this thesis. First, we list the main contributions of this thesis in the area of Web Information System Development and Integration. Next, we describe the conclusion of this thesis based on our study and results as well as the future work.

7.1 Conclusion

This thesis presents a study of the model-driven approach for integrating enterprise and web information systems development. This study aims to seek feasible solutions for meeting HEI needs for proper models and approaches that can serve as standards for WIS development. This thesis provides three key contributions: an IT architecture model, a WIS development method, and WIS evaluation model. We described an IT architecture model in educational institutions was to formulate and how the architecture model influenced the design and implementation of the web information system. We also highlighted selected WIS implementation (by using a case study) and evaluated the case study from the performance and functionality standpoints.

Study results will specifically contribute to the development and integration of existing WISs in the education institutions whereby the WIS that an university plans to establish can answer the needs of the institution. In this context, we can state that the main contributions that we have introduced in each study are the following:

We first study Enterprise Architecture and Model Driven Architecture concepts for formulating an IT architecture model into activities related to Web Information System development to meet HEI needs. Based on an analysis of information technology and information systems of HEI in Indonesia, an architecture model has been proposed according to the situation and needs of national universities. This IT architecture shall guide the institutions in developing an integrated information system to support HEI goals. By applying the IT architecture model, the HEI can evaluate their existing information systems and technologies, while defining the need for a new information system and technology. This architecture model is then applied to University of Lampung as a case study. The outcome is an IT architecture that can be implemented to support integrated WIS development in Unila. The information technology architecture consists of business architecture, information system architecture and technology architecture. The result of this study indicates that the proposed IT architecture model can be a reference for university in order to produce architectures in accordance with the business processes and WIS needs. The definition of a complex architecture model shall help meet all university needs. Nevertheless, universities need not force themselves to create unnecessary business processes, but it would be best for them to focus on areas most suited with their respective needs.

After formulating IT architecture model, in second study we provided WIS development method. In this study, we have adopted and customized MDA-SOA approaches to suit the university's specific needs. We have proposed this method for the development and the integration of WIS for educational institutions. This method defines the WIS development process to automatically obtain web application interface from information system models. This method is then evaluated by applying it in a case study that involves the integration of two existing WIS. Results from this study shed light on how our method can be applied and effectively integrate existing WISs in university. As a case study, one of the proposed WIS in information system architecture of Unila as mentioned in previous study has been implemented by using this method.

We illustrated a development of web application interface by applying our method to integrate two existing WISs at Unila and the results of an evaluation confirmed that the

method can be effectively applied. The implementation of a web application interface (called SIMPEL) has successfully integrated Course Management System into Unila's Academic Information System and work properly. The WIS development process involved in this method allows developers to progress smoothly from the information system models to the generation of the web application. Usefulness of the templates to assistance automatically generate PHP files in the form of MVC files, WSDL files and DDL files is powerful enough to cover the constraints of time effort and write repetitive code. Furthermore, adopted processes and technologies can be reused to develop a different design and create the target web application. The implemented SIMPEL is then analyzed in the teaching/learning activities at Department of Electrical Engineering Unila on trial period in odd and even semester of the 2011/2012 academic year. During this period a total of 887 students were enrolled in 25 courses through the SIMPEL and follow the learning activities in MOODLE. The result showed that SIMPEL can properly perform its functionalities to guide students and teachers in their activities.

This research work is complemented with a new model for evaluating WIS that allows university to assess quality and effectiveness of developed and implemented WIS. We constructed the evaluation model for WIS in the third of our study. This model is built by modifying the concepts of both Technology Acceptance Model (TAM) and IS-Impact due to inadequate comprehensive evaluation for WIS provided by traditional IS evaluation. In view of this, the WIS evaluation model is modified from the aforementioned models by incorporating specific measurements. The evaluation model then applied to assess the quality and effectiveness of WIS development and implementation in Unila. It is described in two parameters: performance evaluation and functionality evaluation. The performance evaluation result of the SIMPLE development concludes that the adoption of WIS development method can help developers efficiently develop and increase productivity by reducing the time effort 2.5 times for web application interface development. The performance evaluation result of the SIMPLE implementation concludes that percentage of unsupported academic and e-learning procedures are zero. It means SIMPEL can cover related academic and e-learning procedures that university need as already defined in requirements specification phase. However percentage of EPSBED data and QA main data unsupported are 30%. In line with this, additional database tables, web services, user interface and functions are necessary in order to

improve the performance. On the other hand, the functionality evaluation of the SIMPLE implementation obtained through questionnaire to 358 users. The evaluation results indicated that 64.7% of users are satisfied with the SIMPEL (highly agree and agree), neutral 30.4% and 4.9% unsatisfied (disagree and highly disagree). It shows that SIMPEL effectively assists the teaching/learning process derived from the elements of assessment : Number of site visits, Enjoyment and Integration. However the result also shows that SIMPEL still need to be complemented and enhanced the quality derived from elements of assessment : Response Time, Accessibility and Reliability. The overall evaluation results show that the evaluation model can be valuable for decision makers in HEIs on evaluating the development and implementation of web information systems.

Three key aspects - information technology architecture model, WIS development method, and WIS evaluation model – incorporated into the WIS education framework research are highlighted in this thesis in contributing towards resolving issues related to the development and integration of existing WISs in educational institutions. This thesis mainly presents a general (national) standard as the reference approach for analyzing, integrating and developing WIS at HEI. It is expected to specifically contribute to the advancement of educational institutions in Indonesia.

7.2 Future Work

The work presented in this thesis provides different possibilities for further work. Whereas some of them concern straightforward extensions of the presented method, there are also possibilities to combine the results of the thesis with other research areas. As the most important and interesting issues the following can be mentioned:

For future work, the model and proposed method in this thesis shall be applied on a much broader scale at our university. Future work includes integrating other web information systems as already planned in Unila's information system architecture. If applied for a longer period, more specific information will be generated for improving and fine-tuning the proposed approach. A complete implementation however has yet to be done. Implementation and migration strategies must be further explored, primarily related to the consequence of using models and approach in a migration project.

To improve the method, we will find other strategies in addressing issues on the integrated WIS and focus on the feasibility of developing a substitute approach to improve and correct information system models prior to their code generation. We need to extend the capabilities of templates, for instance to complete the transformation from PIM to other related files, schemes and diagrams. The web application interface generation strategy can be complemented with the necessary tools for end-users. These tools are developed for the convenience of end-users who may not have programming knowledge. By following instructions on web application development that include directions for appearance format, web services, database and others, users will be provided with adequate information for developing a model as input which will then help generate the web application interface.

To ensure that the integration of existing WISs brings more benefit, we will explore further on rest parts that currently not integrated yet and also the possibility of new functionalities that can be generated from the synchronized systems. An example is the new functionality derived from the WIS integration of Academic Management and Resource Management. This will be significantly useful for data integration and resource sharing with external parties (systems from other universities and the government). An assessment of regulations, policies and documents, such as standard of procedure, therefore becomes vital input for the successful development of WIS. Of equal importance is user feedback and recommendation. Furthermore, research must also involve an evaluation of all stakeholders of the university in order to obtain more comprehensive information for more effective development and better improvements. Apart from information required to identify factors that impact on the acceptance of the information system and quality of WIS being implemented, future work on the quality attributes of WIS also needs to be extended.

The broad variety of systems and their diverse application areas complicates the construction of a web information systems framework. Different system types outside of WIS educational models to a large extent differ in terms of characteristic, structure and user behavior. Many different characteristics of non-functional requirements in WIS development will also affect the overall structure and performance of the web application. However, these differences are not yet fully reflected in our method. We need to improve and evaluate the proposed method to other areas than the educational institutions in our future work.

References

1. <http://evaluasi.pdpt.dikti.go.id/database/pt> (Accessed : July 3, 2013)
2. A. Solihin and Z.A. Hasibuan, "Pemodelan Arsitektur Teknologi Informasi Berbasis Cloud Computing Untuk Institusi Perguruan Tinggi di Indonesia", National Seminar (Semantik 2012), Semarang, Indonesia, 2012.
3. G. Tokdemir, "An Assessment Model for Web-Based Information System Effectiveness", PhD Thesis, Department of Information Systems, METU, Ankara, 2009.
4. S. Jeary, K. Phalp, and J. Vincent, "An evaluation of the utility of web development methods", *Software Quality Control* 17, 2, pp. 125-150, 2009.
5. P.H. Carstensen, L. Vogelsang, "Design of Web-based information systems—new challenges for systems development?" Ninth European conference on information systems (ECIS), pp. 536–547, 2001.
6. C. Gnaho, "Web-based Information Systems Development – A User Centered Engineering Approach", *Lecture Notes in Computer Science*, Springer, 2001.
7. J. Hoick, "4 perspectives on Web information systems", In *Proceedings of the 36th Hawaii International Conference on System Sciences*, IEEE, pp. 1-8, 2003.
8. T. Moritz, K. Schewe, B. Thalheim, "Strategic modelling of web information systems", *International Journal of Web Information Systems*, Vol. 1 Iss: 4, pp. 223 – 240, 2005
9. S. Murugesan, Y. Deshpande, S. Hansen, "Web engineering: A new discipline for development of web-based systems", In *Web Engineering*, volume 2016 of *Lecture Notes in Computer Science*, Springer, pp. 3-13, 2001.
10. R.S. Pressman, "Software engineering. A practitioner's approach", *McCraw-Hill publishing company*, 2000.
11. S. Murugesan, Y. Deshpande "Web engineering: creating a discipline among disciplines," *MultiMedia*, IEEE, vol.8, no.2, pp.82,87, 2001.

References

12. S. M. Beigbeder and C. C. Castro, "An MDA approach for the development of web applications," in ICWE 2004, LNCS, vol. 3140, pp. 300–305, 2004.
13. D. Schwabe, G. Rossi and S. D. Barbosa, "Systematic Hypermedia Application Design with OOHDM," in Proc of the 7th ACM Conference on Hypertext, 1996.
14. O. M. F. D. Troyer and C. J. Leune, "WSDM: A User-Centered Design Method for Web Sites," in Proc of the 7th International World Wide Web Conference, 1998.
15. J. Li, J. Chen and P. Chen, "Modeling Web Application Architecture with UML," in Technology of Object-Oriented Languages and Systems, 2000.
16. G. Booch, J. Rumbaugh, and I. Jacobson, "The Unified Modeling Language: A User Guide", Addison Wesley, 1999.
17. J. Offutt, "Quality Attributes of Web Software Applications," IEEE Software, vol. 19, no. 2, pp. 25–32, 2002.
18. OMG, OMG Model Driven Architecture," Available: <http://www.omg.org/mda>.
19. Mardiana and K. Araki, "A Model Driven Approach for the Interoperability of Web Applications," in Korea-Japan Joint Workshop on ICT, Pohang, 2012.
20. B. Xu, "A Data Secure Solution for Heterogeneous Enterprise Legacy Systems Integration", Journal of Software, 4, 6, pp. 622-629, 2009.
21. S.M. Huang and S.H. Hung et al, "Enterprise Application System Reengineering: A Business Component Approach", Journal of Database Management, 17, 3, pp. 66-91, 2006.
22. W. Y. Mok and J. Fong, "Service-Oriented Architecture, Web Services, XML and Higher Education," in ICHL 2008, Hong Kong, 2008.
23. IEEE Computer Society Coordinating Committee, IEEE Standard Glossary of Software Engineering Terminology, New York: IEEE Standards Board, 1990.
24. Republic of Indonesia, (2005) "Government regulations (PP) No.19 of 2005 of national standard of education" ,Jakarta, Indonesia.
25. National Accreditation Board for Higher Education (Badan Akreditasi Nasional Perguruan Tinggi, BAN-PT) Ministry of Education Indonesia, Book IIIA: Undergraduate program study accreditation, Jakarta, Indonesia, BAN PT, 2008.
26. Directorate General of Higher Education (DGHE), "Dissemination of national higher education database system", Ministry of Education Indonesia, Jakarta, Indonesia, 2010.

References

27. S. Huang; and Y. Fan, "Model Driven and Service Oriented Enterprise Integration---The Method, Framework and Platform," Advanced Language Processing and Web Information Technology, 2007.
28. MOF, "The MOF And OMG's Model Driven Architecture ", <http://www.omg.org/mof/>
29. OMG, "OMG Modeling And Metadata Specifications,"
http://www.omg.org/technology/documents/modeling_spec_catalog.htm
30. D. Lopes, S. Hammoudi, J. Bézivin, F. Jouault, "Mapping specification in MDA: From theory to practice", Book Section, Interoperability of Enterprise Software and Applications, Springer, pp 253-264, 2006.
31. S.J. Mellor and K. Scott et al, "MDA Distilled: Principles of Model-Driven Architecture", Addison Wesley, 2004.
32. S. Sendall and W. Kozaczynski, "Model transformation: the heart and soul of model-driven software development", IEEE, 2003.
33. K. Czarnecki and S. Helsen, "Classification of Model Transformation Approache", Workshop on Generative Techniques in the Context of Model-Driven Architecture, 2003.
34. F. Allilaire, J. Bézivin, F. Jouault and I. Kurtev, "ATL: A model transformation tool," Science of Computer Programming, vol. 72, no. 1-2, pp. 31-39, 2008.
35. "Main Acceleo Site - Acceleo MDA Generator," [Online].
Available: <http://www.acceleo.org/>.
36. "Main Eclipse Site - The Eclipse Open Community," [Online].
Available: <http://www.eclipse.org/>.
37. E. Newcomer, "Understanding Web Services – XML, WSDL, SOAP, and UDDI", Addison-Wesley, 2002 .
38. E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web Services Description Language (WSDL) 1.1, W3C Note": www.w3.org/TR/wSDL, 2001.
39. W. Provost, XML.com, "UML for Web Services":
<http://www.xml.com/lpt/a/ws/2003/08/05/uml.html> , 2003.
40. R. Gronmo, D. Skogan, I. Solheim and J. Oldevik, "Model-driven Web services development," e-Technology, e-Commerce and e-Service, 2004. EEE '04., pp.42,45, 2004.
41. J.A. Zachman, "The Framework of Enterprise Architecture: Background,Description and Utility", www.zifa.com, Oct. 2005.

References

42. J. F. Sowa and J.A. Zachman, "Extending and formalizing the framework for information systems architecture". IBM Systems Journal 31, No. 3, pp. 590-616, 1992.
43. The Open Group, TOGAF Version 9.1, <http://www.opengroup.org/togaf/>
44. S. Leist and G. Zellner, "Evaluation of current architecture frameworks". In Proceedings of the 2006 ACM symposium on Applied computing, 2006.
45. M. Lankhorst, "Enterprise Architecture at Work: Modelling, Communication and Analysis", Second Edition, Springer, 2009.
46. Frankel and S. David, "The Model Driven Architecture: Applying MDA to Enterprise Computing", OMG Press, 2003.
47. Open Group, The Open Group Architecture Framework: Architecture Development Method, 2009.
48. P. Rivett, J. Spencer and F. Waskiewicz, "TOGAF/MDA Mapping, White Paper", The Open Group, 2005.
49. H. Shah and M.E Kourdi, "Frameworks for Enterprise Architecture," IT Professional , vol.9, no.5, pp. 36-41, 2007.
50. M. W. A. Steen, P. Strating, M. M. Lankhorst, H. ter Doest, and M. E. Iacob, "Service-oriented Enterprise Architecture". Service-oriented Software System Engineering: Challenges and Practices, 132, 2005.
51. X. Kong, L. Liu and D. Lowe, "Separation of Concerns: a Web Application Architecture Framework", Journal of Digital Information. Vol. 6, no. 2, 2005.
52. S. S. Ostadzadeh, Aliee and Fereidoon, et al, "An MDA-Based Generic Framework to Address Various Aspects of Enterprise Architecture", Advances in Computer and Information Sciences and Engineering, pp. 455–460, 2008.
53. A. Wegmann, and O.Preiss, "MDA in Enterprise Architecture? The Living System Theory to the Rescue", In Proceedings of the 7th IEEE international Enterprise Distributed Object Computing Conference (EDOC'03), pp. 2-13, 2003.
54. A. Fatolah, S.S. Somé and T. C. Lethbridge, " Enterprise Architecture Using the Zachman Framework: A Model Driven Approach", In Proceedings of the IRMA 2007 International Conference, pp. 65-69 , 2007.

References

55. N. Koch, S.M. Beigbeder, N. Moreno-Vergara and V. Pelechano-Ferragud, "Model-Driven Web Engineering," *European Journal for the Informatics Professional - Joint issue with NOVATICA*, vol. IX, no. 2, pp. 40-45, 2008.
56. P. Cáceres, E. Marcos and B. Vela , "A MDA-Based Approach for Web Information System Development", *Workshop in Software Model Engineering (WISME'03)*, 2003.
57. V. Castro, E. Marcos, P. Cáceres, "A User Service Oriented Method to Model Web Information Systems", *WISE 2004, LNCS 3306*, pp. 41–52, 2004.
58. S. Meliá, and J. Gómez, "Applying Transformations to Model Driven Development of Web Applications ", *ER Workshops 2005, LNCS 3770*, pp. 63-73, 2005.
59. S. Melia, C. Cachero, and J. Gomez, "Using MDA in Web Software Architectures", In *Proceedings 2nd OOPSLA Wsh. Generative Techniques in the Context of Model Driven Architecture*, Anaheim, 2003.
60. S. Ceri, P. Fraternali and A. Bongio, "Web Modeling Language (WebML): a Modeling Language for Designing Web Sites," *Journal Computer Networks*, vol. 33, no. 1-6, pp. 137-157, 2000.
61. D. Distanto, P. Pedone, G. Rossi and G. Canfora, "Model-Driven Development of Web Applications with UWA, MVC and JavaServer Faces," in *ICWE 2007, LNCS 4607*, pp. 457–472, 2007.
62. A.T. Rahmani, V. Rafe, S. Sedighian, and A. Abbaspour, "An MDA-Based Modeling and Design of Service Oriented Architecture," *ICCS 2006, Part III, LNCS 3993*, pp. 578-585, 2006.
63. H. Kim and R.Y. Lee, "MS2Web: Applying MDA and SOA to Web Services," *Soft. Eng., Arti. Intel., Net. & Para./Distri. Comp., SCI 149*, pp. 163–180, 2008.
64. V. De Castro, E. Marcos, and J.M. Vara, "Applying CIM-to-PIM model transformation for the service-oriented development of information systems," *Journal Information and Software Technology*, pp. 87-105, 2010.
65. E. Marcos, V. De Castro, and B. Vela, "Representing web services with UML: a case study," in *Proceedings of the 1st International Conference on Service Oriented Computing, LNCS 2910*, pp. 17–27, 2003.

References

66. V. De Castro, E. Marcos, and M. Lopez-Sanz, "A model driven method for service composition modelling: a case study, " *International Journal of Web Engineering and Technology* 2 , pp. 335–353, 2006.
67. R. Jardim-Goncalves, A. Grilo and A. Steiger-Garcia, "Challenging the interoperability between computers in industry with MDA and SOA," *Computers in Industry*, ScienceDirect, vol. 57, pp. 679–689, 2006.
68. H. K. Kim, "Modeling of Distributed Systems with SOA & MDA," *IAENG International Journal of Computer Science*, IJCS, 35 (4)., vol. 35, no. 4, 2008.
69. M. W. A. Steen, P. Strating, M. M. Lankhorst, H. ter Doest, and M. E. Iacob, "Service-oriented Enterprise Architecture", *Service-oriented Software System Engineering: Challenges and Practices*, 132, 2005.
70. G. Osvalds, "Definition of Enterprise Architecture – Centric Models for The Systems Engineers", TASC Inc, 2001.
71. T. J. Blevins, J. Spencer and F. Waskiewicz, "TOGAF ADM and MDA", The Open Group and OMG, 2004.
72. Directorate General of Higher Education (DGHE), EPSBED, <http://evaluasi.pdpt.dikti.go.id/epsbed> (Accessed : July 3, 2013).
73. V.S. Moertini, "The development of system and information technology facility for universities in Indonesia", General Meeting of the Indonesia Catholic University Association (APTIK), Bandung, Indonesia, 2008.
74. G. F. Nama, "Perancangan Infrastruktur Teknologi Informasi Adaptif Pada Universitas Lampung", thesis magister, MTI, UI, 2012.
75. P. J. Muñoz-Merino, C. D. Kloos and J. F. Naranjo, "Enabling interoperability for LMS educational services," *Computer Standards & Interfaces*, vol. 31, pp. 484–498, 2009.
76. Main PHP Site "PHP" <http://www.php.net/>.
77. Main CodeIgniter Site, "CodeIgniter," <http://codeigniter.com/>.
78. V. Pougatchev and A. B. Kulkarni, "Technical Aspects of the Integrated Management Information System for Educational Institutions," *International Journal of Systems Applications, Engineering & Development*, vol. 5, no. 4, pp. 570-583, 2011.
79. "Main Moodle Site," [Online]. Available: <http://moodle.org/>.
80. <https://moodle.org/stats/> (Accessed : July 3, 2013).

References

81. E. Gutiérrez, M. A. Trenas, J. Ramos, F. Corbera and S. Romero, "A new Moodle module supporting automatic verification of VHDL-based assignments," *Computers & Education*, Elsevier, vol. 54, pp. 562-577, 2010.
82. R. P. Bagozzi, F. D. Davis, P. R. Warshaw, "Development and test of a theory of technological learning and usage.", *Human Relations*, 45(7): 660–686, 1992
83. M.Y. Chuttur, "Overview of the Technology Acceptance Model: Origins, Developments and Future Directions", Indiana University, USA, Sprouts: Working Papers on Information Systems, 2009.
84. G. Gable, D. Sedera, T. Chan, "Re-conceptualizing Information System Success: The IS-Impact Measurement Model", *Journal of the Association for Information Systems*, 9 (7), pp. 377-408, 2008.
85. M. Y. Yi and Y. Hwang. , "Predicting the use of web-based information systems: self-efficacy, enjoyment, learning goal orientation, and the technology acceptance model", *Int. J. Hum.-Comput. Stud.* 59, 431-449, 2003.
86. P. Legris, J. Ingham and P. Colletette, "Why do people use information technology? A critical review of the technology acceptance model", *Information & Management*, 40, pp.191-204, 2003.
87. K. A. Pituch and Y. K. Lee , "The influence of system characteristics on e-learning use. *Computer & Education*", 47, 222-244, 2006.
88. I-F. Liua, M.C Chenb, Y.S Suna, D.Wiblec, C-HKuo, "Extending the TAM model to explore the factors that affect Intention to Use an Online Learning Community", *Computers & Education* 54, pp. 600–610, 2010.
89. Y.-H. Lee, Y.-C. Hsieh and C.-N. Hsu, "Adding Innovation Diffusion Theory to the Technology Acceptance Model: Supporting Employees'Intentions to use E-Learning Systems", *Educational Technology & Society*, 14 (4), 124–137, 2011.

Publications

Journal Papers

- [1]. Mardiana and Keijiro Araki, "EA-MDA Model to Resolve IS Characteristic Problems in Educational Institution", The International Journal of Software Engineering & Applications (IJSEA), Vol.4, No.3, pp. 1-20, May 2013.

International Conference Papers

- [1]. Mardiana, Keijiro Araki and Yoichi Omori, "MDA and SOA Approach to Development of Web Application Interface," in Proceedings of 2011 IEEE Region 10 Conference (TENCON 2011), Bali, Indonesia, pp.226-231, Nov 2011.
- [2]. Mardiana and Keijiro Araki, "SIMPEL: An Innovative Web Application Interface Supporting Online Course Management System," in Proceedings of 11th IEEE International Conference on Information Technology Based Higher Education and Training (ITHET 2012), Istanbul, Turkey, pp.212-219, June 2012.
- [3]. Mardiana and Keijiro Araki, "A Model Driven Approach for the Interoperability of Web Applications," in Proceedings of Korea-Japan Joint Workshop on ICT, IPSJ&KIISE, Pohang, Korea, pp.57-60, Sept 2012.

Index

Academic management	52, 59, 60, 88	Entity-Relationship	14
Acceleo	21, 68, 71, 73	EPSBED	40, 44, 50, 52, 127
Active Server Pages	14	Evaluation model	5, 120, 129
AJAX	14	Executive Information System	45, 53
Application Architecture	25, 43, 51, 54	EXtensible Markup Language	15
Architecture Development Method	27	Grade Calculation	93, 98
ATL	21	Higher Education Institution	1, 4, 16, 33, 45, 149
Business Architecture	24, 27, 42, 49, 60, 65	Individual Impact	121, 122, 123
CIM	18, 47	Information Effectiveness	122, 123, 124
Code Generator	79	Information Quality	121, 122, 123
Codeigniter	80, 110	IS	1, 2, 3, 4, 5, 7, 8
Common Warehouse Metamodel	19, 148	IS Architecture	24, 27, 42, 50, 60, 65
Community Relationship Management	52, 53, 60	IS-Impact	9, 119, 120, 121, 128
Course Enrolment	96, 102, 109	IT	1, 24, 41
Course Management Systems	89	IT Architecture Model	24, 41
Course Registration	96, 103	IT Services	52, 53, 60
Critical Success Factors	52, 120	Java Server Pages	14
CRUD	31, 32	Knowledge management	52, 59, 60
Data Architecture	25, 42, 50, 51	Learning Strategist	97, 103
DDL	8, 109	LOC	125
DDL Templates	72, 85	Local Area Network	55
Disaster Recovery Plan	56	Measurement Dimensions	123
Educational institutions	1, 12, 59	Meta Object Facility	19
E-learning	44, 54, 89, 91, 96, 112, 127, 130	Metamodel	19, 20, 69
Enterprise Architecture	2, 5, 7, 11, 24, 25, 27, 28, 29, 42, 44	Model Driven Architecture	2, 5, 7, 11, 14, 17, 18, 44
		Model View Controller	8, 30

Index

MOODLE	89	System Quality	121, 122, 123
MVC Templates	72, 85	Technology Acceptance Model	119, 120, 121
National Higher Education Database		Technology Architecture	24, 25, 27, 41, 44, 55, 60
PDPT	17, 40	Technology Infrastructure	47
OMG	18	Templates	70
OOHDM	14	TOGAF	25
Organisation Capabilities	122, 123	Triple Principle of HEIs	1, 33, 45
Organizational Impact	121, 122, 123	Ubiquitous Web Application	30
Perceived Ease of Use	121, 122	UDDI	22
Perceived Usefulness	121, 122	Unila	8, 35, 49, 60, 87
PHP	14	User Capabilities	122, 123
PIM	18, 48	User Registration	95
PSM	18, 48	UWE	14
Resource management	52, 60	Web Application Interface	62
Service Oriented Architecture	5, 7, 11, 17	Web Information System	2
SIAKAD	89, 90, 91	Web Modeling Language	14
SIMPEL Development	91	Web Service Description Language	8, 22
Single sign-on	53, 55	WIS development method	5, 6, 8, 67, 85
SOAP	22	WIS Development Process	68
SODM	31	WSDL	8, 22
Standard Query Language	66	WSDL Templates	72, 85
Support Activities	44, 52		
SWOT	52		
System Effectiveness	122, 123, 124		

List of Abbreviations

ADM	Architecture Development Method
AJAX	Asynchronous JavaScript and the extensible Mark-up Language
API	Application Program Interface
ASP	Active Server Pages
ATL	ATLAS Transformation Language
BPMN	Business Process Modelling Notation
CIM	Computation Independent Model
CRM	Community Relationship Management
CRUD	Create, Retrieve, Update, Delete
CSF	Critical Success Factors
CSM	Course Management Systems
CWM	Common Warehouse Metamodel
DDL	Data Definition Language
DGHE	Directorate General of Higher Education
DRP	Disaster Recovery Plan
EA	Enterprise Architecture
EIS	Executive Information System
EPSBED	Evaluasi Program Studi Berbasis Evaluasi Diri (Study Program Evaluation Based on Self-assessment)
ER	Entity-Relationship
FP	Function Point
GLOC	Generated Lines of Code
GUI	Graphical User Interface

List of Abbreviations

HEI	Higher Education Institution
HLOC	Handwritten Lines of Code
HTTP	Hypertext Transfer Protocol
IE	Information Effectiveness
II	Individual Impact
IQ	Information Quality
IS	Information system
IT	Information Technology
JSP	Java Server Pages
LAN	Local Area Network
LOC	Lines of Code
MDA	Model Driven Architecture
MOF	Meta Object Facility
MOODLE	Modular Object-Oriented Dynamic Learning Environment
MS2Web	MDA and SOA to web services
MVC	Model View Controller
OC	Organisation Capabilities
OI	Organizational Impact
OMG	Object Management Group
OOHDM	Object Oriented Hypermedia Design Model
PDPT	Pangkalan Data Pendidikan Tinggi (National Higher Education Database)
PEU	Perceived Ease of Use
PHP	PHP: Hypertext Preprocessor
PIM	Platform Independent Model
PSM	Platform Specific Model
PU	Perceived Usefulness
QA	Quality Assurance
SE	System Effectiveness
SIKAD	Sistem Informasi Akademik (Academic Information System)
SIMPEL	Sistem Informasi Pembelajaran E-learning (Web Application Interface)
SOA	Service Oriented Architecture

List of Abbreviations

SOAP	Simple Object Access Protocol
SODM	Service-Oriented Development Method
SQ	System Quality
SQL	Standard Query Language
SSO	Single sign-on
SWOT	Strengths, Weaknesses, Opportunities, and Threats
TAM	Technology Acceptance Model
THLOC	Total Handwritten Lines of Code
TOGAF	The Open Group Architecture Framework
UC	User Capabilities
UDDI	Universal Description, Discovery and Integration
UML	Unified Modeling Language
UNILA	University of Lampung
UWA	Ubiquitous Web Application
UWE	UML-based Web Engineering
WebML	Web Modeling Language
WebSA	Web Software Architecture
WIS	Web Information System
WSDL	Web Service Description Language
WSDM	Web Site Design Method
WWW	World Wide Web
XML	EXtensible Markup Language