A Model Driven Approach for the Interoperability of Web Applications

MARDIANA[†1] KEIJIRO ARAKI[†2]

Abstract: Along with advances in information technology today the need for an integrated information system absolutely must be done in any institution, including educational institutions. MDA and SOA approach has been introduced and used in some research for the integration solution. In this work, we used this approach to develop a web application that acts as an interface and integrate the existing web information systems. We described the process followed to implement the approach and a set of models at the different abstraction levels include the model transformations among them. It can be starting from a high level business model and gained a low level form. For a case study, we applied the approach to develop a web application interface (called SIMPEL) as an interface between two existing applications in University of Lampung (Unila).

Keywords: Interoperability, web application interface, MDA,SOA, MVC, metamodel

1. Introduction

New technologies are evolving very rapidly, especially in the field of web development requires the companies or institutions considered the possibility of changes in technology used or integrate the existing systems. In educational institutions, technology and information systems used are very diverse. Mostly web-based information systems are adopted to support the academic processes. Unfortunately, they are not able to share resources and operate with one another. In this case interoperability is very essential in order to be able to realize integration of many information systems. By this integration, it is expected that the user will find easier in data access and management. Also, the decision making is expected to be more accurate and timely because information can be obtained immediately [1]. The effectiveness and efficiency of a service can be achieved by optimizing available information in any existing information systems. This can give positive impacts to the university's efforts to provide the highest levels of service to students, teachers, and other constituents.

Considering this background, a web application as an interface is required to enable the various information systems by using web service technology. Currently developing web applications for integration purposes has become more popular due to the increasing number of available web services. A web application that is intended in this paper is also an information system which provides the facility to access complex data and interactive services to users through the web. This web application interface benefit is to create integration between existing information systems within the local university and also feasible to improve to a wider scale in the field of education. It must be flexible, prevents the dependency and relatively easy for future modification and revision. In order to support the management's work, this application essential to provide the facilities needed for decision making, strategic planning and control. The other requirements needed to become a reliable system; improve the communication, self-service functionality, content management and publishing, reporting and statistics, information extraction, as well as data security and privacy [2].

Developing a new web application as an interface is preferred rather than modifying existing applications. Modification is not the approach followed because the developers may not comprehend the existing applications development. Even if, the applications are based on open source, which allows for modification, it can generate further problems. For example, problems may arise when operate the upgrade version. Changes of version may cause the modified application to be incompatible with previous versions. Furthermore, the solution obtained on modification only for the problem addressed cannot apply in general problem. Developing new applications can prevent problems arising due to changes in existing applications as well as provide several new functionalities.

This work describes a development of web application interface applying the benefits of Model Driven Architecture (MDA) [3] and Service Oriented Architecture (SOA) [4]. The modeling from the perspective of how services are used in web application system can be developed by using this approach. We investigate the functional requirements of system by identifying services from the basic elements in existing systems. The defined integration processes and services from previous models can be transformed into the next required models with the specific platform. In terms of integration of information systems, the advantage of using MDA is that this approach separates the different models at different levels. The separation and the different level of models will increase the flexibility to handle the complexity of the process. As technology changes, the model at the level of the CIM (Computation Independent Models) and PIM (Platform Independent Models) models do not need to be changed. Only at the level of PSM (Platform Specific Models) model should be changed and it will be simpler and cheaper than having to change the overall software.

The rest of the paper is organized as follows: section 2 presents initial motivation and related work. Section 3 gives web application interface development. How to apply MDA to

[†1] Kyushu University
[†2] Kyushu University

development process are described in section 4. Section 5 presents result and evaluation. Finally, our concluding remarks and future work are presented in section 6.

2. Initial motivation and related work

The demand to integrate existing information systems in university stems from the necessity to combine data throughout the departments, faculties or other work divisions as well as to use the data for different purposes. By the integration, some data which were previously entered in one system is also available in other systems automatically. The users should not make a transaction twice while using the systems because the actions initiated in one system will also affect others directly. The development of integrated information system is also considered by University of Lampung (Unila, Indonesia) due to the number of existing information systems with different platforms and database. Unila has information systems which developed by using own source-based development, open source-based development, and for some special purpose the application also already developed by the government and ready to be implemented. 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. Therefore, the integration of the system is required to enhance the effectiveness and efficiency.

Model driven approach for interoperability means interoperability requirements are added to system models and from the models the integrated system are directly generated. In order to produce a feasible web application interface by using this approach, several key questions have to be answered such as: how to align the new system with running systems and strategic plan of the university? What basic services include in integration platform? How can web application modeling be viewed from different level? How to adopt web services and software reusability? What methods are used to abstract the complexity of business processes? How to utilize the available technology and architecture for developing platform-independent models to generate code automatically?. In our work we specify several answers to those questions systematically, and then became the basis for the requirements to develop the web application interface. We design a modeling approach that is specific to integrate the existing information systems in educational field. As a case study we apply the models to a web application interface (called SIMPEL) as an interface between two existing systems in Unila. The existing systems are the SIAKAD (Academic Information System) and the MOODLE (Modular Object-Oriented Dynamic Learning Environment) as an example of Course Management Systems (CMS).

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. Some technologies used to implement SOA are Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI). They are all the main techniques for data exchange, data integration and data sharing between existing systems. The MDA at this time has been widely proposed for simplification and automation in the field of developing web applications. It upgrades the quality and speed of developing web applications, as well as their maintenance. A powerful point of the MDA is the utility of generate automated tools and services to describe the abstraction models and also facilitate the transformation between models. There is already some existing research used model driven for the development of web applications. For example author in [5] show how MDA is applied to support the web development process by defining model and metamodels. In [6] a web model driven approach called WebSA (Web Software Architecture) is presented by using UML profiles. They provide a set of web specific models form each process to the implementation phase. Some approaches also adopt the MVC architecture pattern for its PIM such as in [7] and [8]. The authors in [7] provide WebML (Web Modeling Language) meanwhile [8] proposed UML-MVC logical model as an intermediate model to be used between UWA (Ubiquitous Web Application) conceptual design and the implementation phase. However, the research that present this approach for integrating information systems by defining MVC models and templates for specific purposes such as for educational purposes rather view. We describe this approach with the educational purpose in this work.

3. Web application interface development

3.1 Web Services as integration path

The initial step should be done for developing the web application interface is the identification of business processes and system capabilities of each existing system. Redundant processes and information should be discharged to avoid duplicate input data, divergence in the information flow or inconsistent processes. Next step is mapping the data and information from the documentation 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. Afterwards, 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.

3.2 Web application design

The process and technologies used to develop the application can be reused to develop application for different target platform. The application design in this case is shown in figure 1. It is designed following the MVC pattern [9]. Controller will handle the user’s request. Then it calls to web service clients, selects and covers the response views. Web services provided by existing applications.

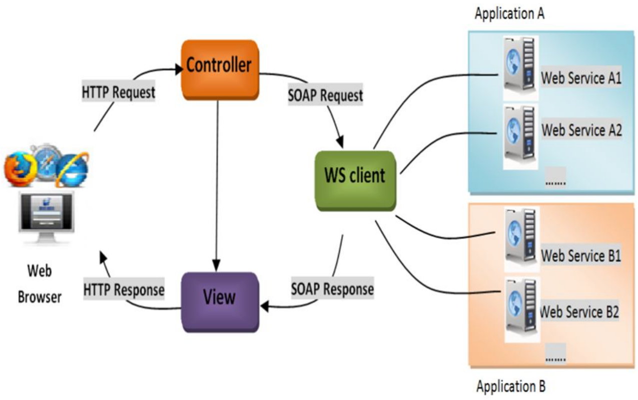


Figure 1. Web application design

4. Apply MDA to development process

The web application development process is based on the MDA concept. During the design phase, modeling is used to define requirements, provide models detail in various levels. Then, the model transformation is performed with transformation rules. The models contribute to automatic code generation. Our approach will describe a set of models at the different abstraction levels and also model transformations among them. It can be starting from a high level business model and gained a low level form. All the development process is comprised of the five main steps below:

- *Creating business models.*

These models are used to describe the environment in which the system will be used, with no direct orientation on how it will be implemented. It can specify the requirements, use cases and the main flow of the system from the customer view. In this step we also define a metamodel for representing the design of web application.

- *Creating process models.*

These models are used to model the process and the structure of the system, independently from the technological details that will be implemented. The processes are captured by UML activity diagram and structures are capture by UML class diagram. The models are then allowed to modified or transformed into a PSM that express the specific implementation. In this case, we adopt MVC pattern as a target technology for the generated web application.

- *Defining templates*.

In order to generate the web application automatically, model transformations, so-called templates, is required to define. Transformation rules to transform a given model to model or text are defined by the templates. The template responsible for implementing class for the Controller of the MVC pattern those are responsible for receiving a request, invoke the model to perform the requested operations and send the data to the View. The View formats the data to be presented in a web application as a PHP file output.

- *Code generation.*

By using models and defined templates, code generation step is executed and automatically generates the source code. Generally source code can be generated to the several purposed of platforms, however in this work we concentrate to the web applications purposes using the PHP code. We utilized Acceleo for creating the templates and generating PHP-based web application.

- *Web Application Source Code.*

This application is implemented using the scripting language PHP and additional technologies such as Apache web server and MySQL database server as the target environment for deployment. Application using CodeIgniter PHP framework follows MVC pattern. It is also expressed by the directory structure. Figure 2 shows the MVC pattern that used to illustrate the structure of web application.

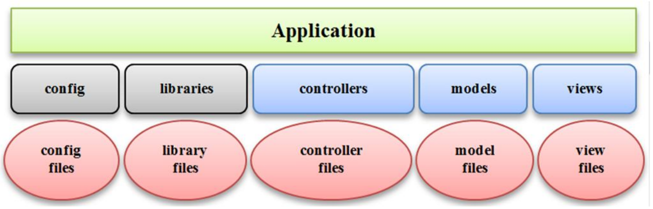


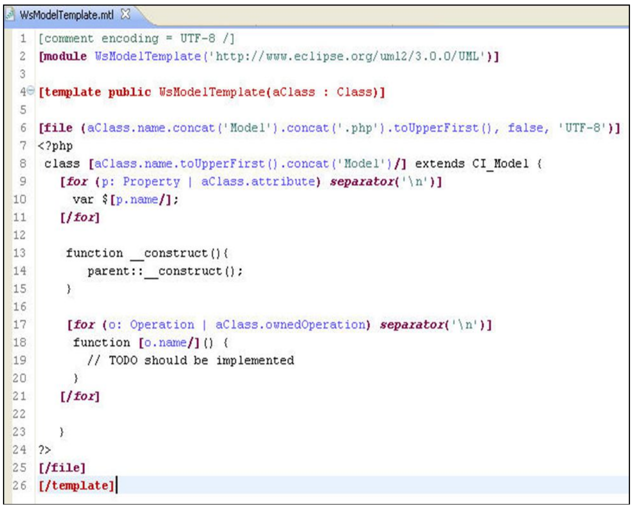
Figure 2. MVC pattern of web application

5. Result and evaluation

In this work, we designed a set of models at the different abstraction levels in the context of web application interface. A specific metamodel acquaintance of the UML metamodel created for the modeling of this concern. Based on the result obtained from the modeling process, we have generated artifacts that can be used to create concrete web pages needed. However the modeling is only half of the entire process of MDA. The model transformations, so-called templates, to generate the target system are equally important. The usefulness of the template is to define a model transformation rule that transform a given model to another model. We defined templates to generate a given modeled that can be deployed on the Apache server. A developer of web application interface can choose according to the design and the data want to be displayed. The templates are responsible for implementing classes for the MVC pattern which consists of templates for the Model, View and Controller implementation.

CrudModelTemplate and *WsModelTemplate* are the template used for the Model template. Both of these template functions for operations *Insert*, *Update*, *Delete*, *List* data from the database and web services respectively. The *ControllerTemplate* as a defined template for the controller serves to create object models and call the function models. Meanwhile to adopt the requirements of views of the web application interface, we defined six View templates. Templates are as follows: *CrudViewTemplate* for CRUD (Create, Read, Update, Delete) database operations, *MenusViewTemplate* for menu generation, *TreeListViewTemplate* for view options ; tree of data view on the right, list of data view and form on the left side, *ListViewTemplate* only for a list of data view option, *LoginViewTemplate* for displaying a login user form, *MainLayoutViewTemplate* for layout and the last *DetailViewTemplate*, to display detailed data such detailed data user or a confirmation page. Figure 3 shows the example of WsModelTemplate that can be used for generating MVC model.

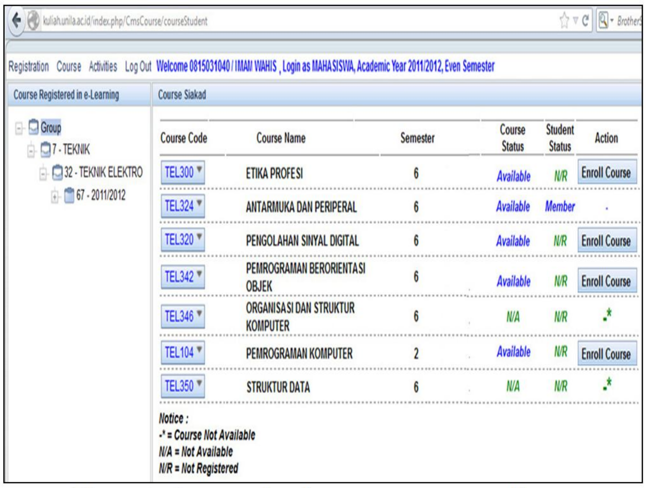
This template will generate of properties and functions from a number of class diagram of MVC model.



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

[template public WsModelTemplate(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) separator('\n')]
     var $[p.name/];
   [/for]

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

    [for (o: Operation | aClass.ownedOperation) separator('\n')]
     function [o.name/]() {
       // TODO should be implemented
     }
   [/for]

   }
?>
[/file]
[/template]
```

Figure 3. WsModelTemplate

An example of generated web page can be seen in figure 4. This figure shows the SIMPLE page feature on the course enrolment that can be used by the students to enrol the courses in the MOODLE based on their study plan that existed in the SIAKAD. On the right side, there is a view of the tree of the data group which consists of the faculty, department, and the academic year. On the left side, there is a list of data that show the student's study plan, course status, student status and a button to enroll a course. The list of data for this course enrolment displays data obtained from the SIAKAD web service (*course_service*) and the MOODLE web services (*get_course, get_categories, add_course, add_student and affect_user_to_course*).



Welcome 0815031040 / IMAM WAHIS , Login as MAHASISWA, Academic Year 2011/2012, Even Semester

Course Code	Course Name	Semester	Course Status	Student Status	Action
TEL300	ETIKA PROFESI	6	Available	N/R	Enroll Course
TEL324	ANTARMUKA DAN PERIPERAL	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 4. Generated web page to course enrollment

All generated PHP files are in form Model, View, Controller in the appropriate folder with MVC pattern as shown in figure 2. These PHP files are producing the set of web pages in order to integrate the SIMPLE and the MOODLE.

We evaluate our approach by evaluating the meta-model, the defined templates, as well as the handled generated code and handwritten code. We also evaluate whether the generated PHP files accordance with the requirements specified in the previous model. The result gives an overview how our approach can be extended and improved by extending and improving the defined meta-model, the created templates for the generation of the modeled web application interface.

6. Conclusions

We have presented a development of web application interface by applying the MDA-SOA approach to integrate two existing information systems. A case study conducted by designing web application interface (SIMPEL) as an interface between academic information systems and CMS at Unila using this approach. The results have shown that this approach can be applied and work properly. It allows developers go from requirements specification at the beginning to generate web application automatically. It can also assist in the development of web application that invokes web services to enable the interoperability of two existing web application. From the developer viewpoint, by using this approach can help developers efficiently develop and minimize the time-consuming process of developing their web application interface. Additionally, the process and the technologies adopted is expected can be reused to develop a different design and the target platform. Our plan for future work is consider about another approach for proving the correctness of models given before it will be generated.

Reference

1. 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.
2. T. Kudrass, "Integrated University Information Systems," in Eight International Conference on Enterprise Information Systems, Databases and Information Systems Integration, 2006.
3. "OMG Model Driven Architecture," [Online]. Available: http://www.omg.org/mda.
4. T. Erl, Service-Oriented Architecture - Concepts, Technology, and Design, Prentice Hall, 2005.
5. N. Koch, S. Melia-Beigbeder and J. Vara-Mesa, "Model-Driven Web Engineering", European Journal for the Informatics Professional - Joint issue with NOVATICA, IX(2), pp. 40–45, 2008.
6. S. Meliá-Beigbeder and C. C. Castro, "An MDA approach for the development of web applications", in ICWE 2004, LNCS, vol. 3140, pp. 300–305, 2004.
7. S.Ceri, P. Fraternali, A. Bongio, "Web Modeling Language (WebML): a Modeling Language for Designing Web Sites", Journal Computer Networks 33(1-6), pp. 137-157, 2000.
8. D. Distante, 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.
9. S. Brown, S. Dalton, D. Jepp, D. Johnson, S. Li and M. Raible, "Web-Application Design," in Pro JSP 2 , Apress, 2005, pp. 535-569