Component Based Development Using the PRADO Framework

Zaitun A. B. and Vanisri Ramasamy
Department of Information Science,
Faculty of Computer Science and Information Technology,
University of Malaya,
Kuala Lumpur,
Malaysia.

Tel no: +603 7976432
Fax. No: +603 7579249
zab@um.edu.my

ABSTRACT

Organizations require efficient information systems to support their daily operations in order to remain competitive and have a cutting edge over their contenders. These information systems need to be ready at a fraction of the time given to systems development work in the pre-Internet era. One of the numerous approaches to cut development time is software reuse. In this paper, we present to the readers our findings and experience in developing a web-based system using component-based development with the PRADO framework.

In this paper, we will be focusing on software reusability concept using component-based software engineering, design patterns and web application frameworks and identify the approach to be used to develop the Postgraduate Student Administration System (PSAS). We will examine a few of the available free web application framework, which focuses on the open source technology, analyze the advantage and identify the web application framework to develop PSAS – a system that can be implemented in academic institutions to administer the postgraduate student's dissertation data.

Keywords: Component Based Development, Application Framework, Software reusability

1. Introduction

The Internet has changed the way we conduct our business and the way we communicate. Business and daily communications now moved at a very much faster pace. In order to remain competitive, business organizations need the support of efficient and flexible web-based information systems. Therefore there is a need for organizations to develop these web applications quickly in order to meet customers' requirements. There is also a need to reduce the process risk and increase the reliability of these web applications. These requirements can be fulfilled by adopting the software reusability approach to system development. Reuse based software engineering is an approach that tries to maximize the reuse of existing software. According to Boehm (1984), software reuse will be one of the major sources of saving in software development over the next 15-20 years. By reusing systems or system parts that already have been developed, an organization enhances its possibilities to both improve the productivity and the quality of the produced software. Since software reuse costs are often higher than expected, several approaches to development with reuse have been proposed. Sommerville (2004), pointed

out that component-based or component-based software engineering, design patterns and application framework will be useful techniques in implementing software reusability. Web application developers can use the approaches to develop a robust and high quality web application quickly.

2. Application Framework

Wirfs-Brock and Johnson (1990) defined a framework as a sub-system design made up of a collection of abstract and concrete classes and the interface between them. Particular details of the application sub-system are implemented by adding components and by providing concrete implementations of abstract classes in the framework. Frameworks are rarely applications on their own. Applications are constructed by integrating a number of frameworks. A framework is a generic structure that can be extended to create a more specific sub-system or application. It is implemented as a collection of concrete and abstract object classes.

An application framework actually refers to a set of libraries or classes that are used to implement the standard structure of an application for a specific operating system. By bundling a large amount of reusable code into a framework, much time is saved for the developer, since the task of rewriting large amounts of standard code for each new application that is developed is saved. Application frameworks became popular with the rise of the graphical user interface (GUI), since these tended to promote a standard structure for applications. It is also much simpler to create automatic GUI creation tools when a standard framework is used, since the underlying code structure of the application is known in advance. Frameworks are often instantiations of a number of patterns. Object-oriented programming techniques are usually used to implement frameworks such that the unique parts of an application can simply inherit from pre-existing classes in the framework.

Application Frameworks and Components

Components provide the developer with a group of classes and abstractions to inherit from or instantiate them. Unlike frameworks, components are more general and are not tied to a specific application domain.

Frameworks use components and extend their benefits in two ways:

1. By using a framework, not only the classes are instantiated or inherited but also the underlying architecture.
2. Frameworks are active and exhibit "inversion of control" at run-time. Inversion of control, one of the primary benefits of frameworks.

Application Frameworks and Design Patterns

During the development process of complex software systems, developers may face some programming problems. Some of these problems repeatedly appear in their next software projects. Until mid-1990s, the solutions of these problems were located only in the minds of expert developers. This is not the ideal location. The documentation of these problems and their proven solution is now known as "design patterns." Patterns support the reuse of design expertise by articulating the static and dynamic aspects of successful solutions to problems that arise when building software in a particular context. Patterns guide framework design and use. Patterns can be viewed as more abstract micro-architectural elements of frameworks that document and motivate the semantics of frameworks in an effective way.

Web Application Framework

A framework is an extensible structure for describing a set of concepts, methods, technologies, and cultural changes necessary

for a complete product design and manufacturing process. Frameworks provide a mechanism that guides users through a proper order of steps, applications, and data conversions via a common interface to the process being followed. A framework consists of a set of super classes, APIs, and other libraries appropriate for solving a particular domain of problems (in this case, developing a Web application) (Package Aquarium, 2005)

Choosing a Web Application Framework

When choosing a framework we should consider answering the following question:

- Does the framework handle most things that are common to applications of the kind you wish to develop?
- Does the framework have a strong user community to back it?
- How much does the framework cost?
- How steep is the learning curve for the framework?

The cost is an important factor—although "free" in most cases also means widespread usage, community support, and no dependence on a single vendor. It takes some time for developers to get used to a framework and be good at it. So choose a framework as a long-term strategy. You cannot be switching frameworks for every project. Sticking with one framework also helps as once the expertise in that framework builds up; customizing the framework also becomes a possibility.

A tempting option is for organizations to build their own framework to address needs specific to the kind of work the organization undertakes. Thoroughly testing and maintaining a framework is a huge task that will need dedicated people resources. These few people become critical and are the only source of support for others using that framework.

There are many free frameworks available to choose from. Some of the web application frameworks that have been evaluated for this dissertation are based on open-source web application frameworks in PHP. Those frameworks are as below:

a. php.MVC

php.MVC (Wildenauer, 2006) implements the Model-View-Controller (MVC) design pattern, and encourages application design based on the Model 2 paradigm. This design model allows the Web page or other contents (View) to be mostly separated from the internal application code (Controller/Model), making it easier for designers and programmers to focus on their respective areas of expertise. The framework provides a single entry point Controller. The Controller is responsible for allocating HTTP requests to the appropriate Action handler (Model) based on configuration mappings. The Model contains the business logic for the application. The Controller then forwards the request to the appropriate View component, which is usually implemented using a combination of HTML with PHP tags in the form of templates. The resulting contents are returned to the client browser, or via another protocol such as SMTP. php.MVC is a PHP port of Jakarta Struts. It currently supports many features of Struts, including declarative application configuration via the XML digester. For example, mappings from the various Action business logic components to appropriate results pages can be specified declaratively in the XML configuration file.

b. binarycloud

binarycloud (Turing, 2006) is a web application framework for the PHP language. It provides a set of services that are frequently used when writing web applications and helps to improve reusability by providing a modular application infrastructure. The centrepiece of this framework is the presentation system (Node Tree), which enforces a strict separation of

your business logic from your presentation logic. The Node tree uses a unique, hierarchical scale free object design that is extremely powerful because applications can be arbitrarily infinitely nested. The Smarty template engine is the default rendering layer and is tightly integrated into binarycloud.

c. Achievo ATK

Achievo ATK (ATK5, 2006) is an object oriented Web Application Framework, written in PHP. It is targeted at developers who wish to focus on business logic, instead of coding HTML. Where other application frameworks mainly provide a large set of utility classes, ATK provides a complete framework that requires only small amounts of code to get usable applications, while maintaining full flexibility. In other words, even 10 lines of code get you a working application, but everything generated for you, can be 100% customized. It is Achievo ATK's belief that will revolutionize the way business applications are built. Therefore ATK is referred as a 'business framework' or 'enterprise framework'.

d. Smarty

Although Smarty is known as a "Template Engine" (Smarty Template Engine, 2006), it would be more accurately described as a "Template/Presentation Framework." That is, it provides the programmer and template designer with a wealth of tools to automate tasks commonly dealt with at the presentation layer of an application. Smarty is not a simple tag-replacing template engine. Although it can be used for such a simple purpose, its focus is on quick and painless development and deployment of application, while maintaining high-performance, scalability, security and future growth.

e. PRADO

PRADO (PRADO Group, 2006), is a component-based and event-driven Web programming framework for PHP 5. PRADO re-conceptualizes Web application development in terms of components, events and properties instead of procedures, URLs and query parameters. A PRADO component is a combination of a specification file (in XML), an HTML template and a PHP class. PRADO components are combined together to form larger components or complete PRADO pages. Developing PRADO Web applications mainly involves instantiating pre-built and application-specific component types, configuring them by setting their properties, responding to their events by writing handler functions, and composing them into application tasks.

The benefits that PRADO provides to Web application developers are as below developers:

- **Reusability** - Codes following the PRADO component protocol are highly reusable. Everything in PRADO is a reusable component.
- **Ease of use** - Creating and using components are extremely easy. Usually they simply involve configuring component properties.
- **Robustness** - PRADO frees developers from writing boring, buggy code. They code in terms of objects, methods and properties, instead of URLs and query parameters. The latest PHP5 exception mechanism is exploited that enables line-precise error reporting.
- **Performance** - PRADO uses a cache technique to ensure the performance of applications based on it. The performance is in fact comparable to those based on commonly used template engines.
- **Team integration** - PRADO enables separation of content and presentation. Components, typically pages, have their content (logic) and presentation stored in different files.

PRADO focuses more on establishing a standard of reusing codes and event-driven programming. Experienced Windows programmer using Visual Basic or Delphi will find Web programming with PRADO is very similar. Most of the time, the only thing needed to be done is to set component properties and respond to component events. Higher level of code reusability can be achieved based on PRADO components.

3. The Development of PSAS

Academic institutions are currently faced with the challenging problem of managing as well as administrating student data, which is growing every day. The higher the student level the more complicated is his/her information profile. The task of keeping track of a post-graduate student who is doing research under a lecturer is often so cumbersome that departments run shadow spreadsheets often resulting in inaccurate data. In a wider context, in an educational environment which is supported by different department, the absence of a universal means of identifying students resulted in using different enrollment networks and not having the capacity to transfer student records across department makes the data management process difficult. Therefore, an online database system, which is capable of integrating to the main student database and display as well as manages the student's research progress, is badly needed.

The focus of this project is to provide such integration through an object model for student record. This object model will be accessed through a web gateway to a meta-directory, a virtual database that abstracts all information. Parallel to the development and prototyping of the object model is the design and development of a website by utilizing the software reusability concept. Web application framework is the approach identified to utilize the software reusability concept in developing the PSAS system.

The system is realized through the PRADO's 3-Tier architecture as shown in Figure 1. The figure describes that the system consists of presentation layer, service layer and data access layer. Designing application in layers, or tiers, is useful for many different reasons. Efficient layering can give structure to the application, promote scalability, and ease long-term maintenance requirements for code. The advantages of 3-Tier architecture are as below:-

- Clear separation of user-interface-control and data presentation from application-logic. Through this separation more clients are able to have access to a wide variety of server applications. The two main advantages for client-applications are clear: quicker development through the reuse of pre-built business-logic components and a shorter test phase, because the server-components have already been tested.
- Re-definition of the storage strategy won't influence the clients. RDBMS' offer a certain independence from storage details for the clients. However, cases like changing table attributes make it necessary to adapt the client's application. In the future, even radical changes, like let's say switching form an RDBMS to an OODBS, won't influence the client. In well designed systems, the client still accesses data over a stable and well designed interface which encapsulates all the storage details.
- Business-objects and data storage should be brought as close together as possible, ideally they should be together physically on the same server. This way - especially with complex accesses - network load is eliminated. The client only receives the results of a calculation - through the business-object, of course.
- In contrast to the 2-tier model, where only data is accessible to the public,

business-objects can place applications-logic or "services" on the net. As an example, an inventory number has a "test-digit", and the calculation of that digit can be made available on the server.

- As a rule servers are "trusted" systems. Their authorization is simpler than that of thousands of "untrusted" client-PCs. Data protection and security is simpler to obtain. Therefore it makes sense to run critical business processes that work with security sensitive data, on the server.
- Dynamic load balancing: if bottlenecks in terms of performance occur, the server process can be moved to other servers at runtime.
- Change management: of course it's easy - and faster - to exchange a component on the server than to furnish numerous PCs with new program versions. It is, however, compulsory that interfaces remain stable and that old client versions are still compatible. In addition such components require a high standard of quality control. This is because low quality components can, at worst, endanger the functions of a whole set of client applications. At best, they will still irritate the systems operator.

Figure 1 below shows the system architecture.

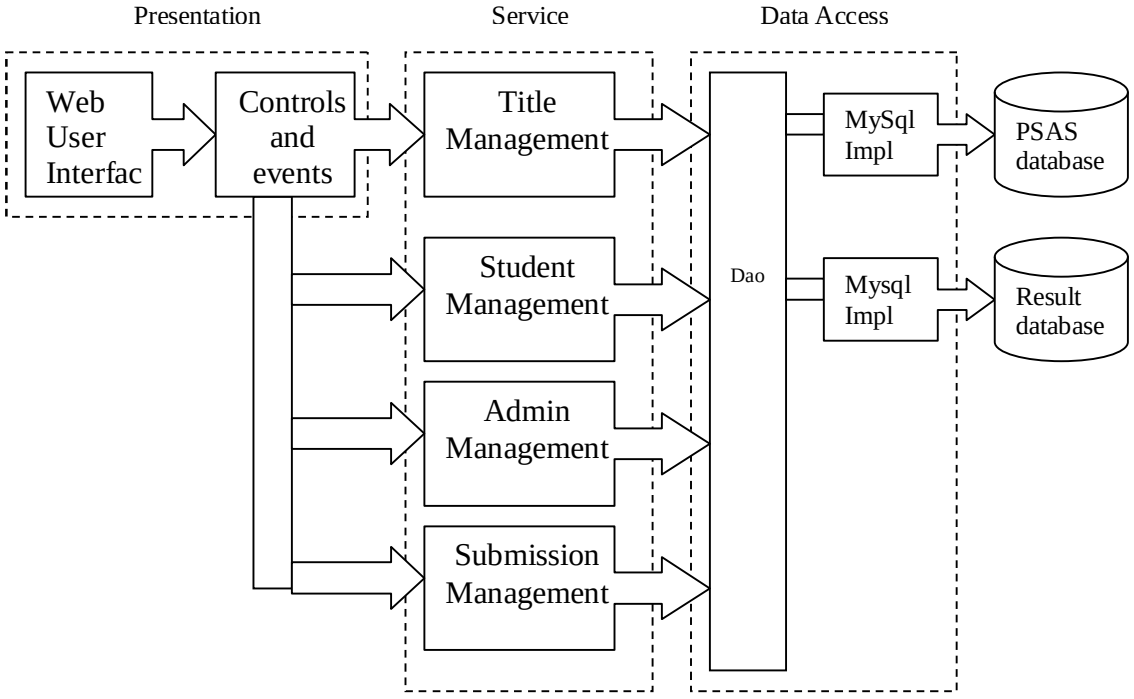

Figure 1: PSAS System Architecture

4. System Implementation

The system is implemented using the three layered architecture which are the presentation (*.tpl), service (*.php), and data access (*.php). Apart from this, every module will have an application specification files (*.spec). The classes to be inherited from the framework will be identified and the coding of the hotspot and the data module as well as the user interface will be done in this stage. The following files were created to demonstrate the application of the PRADO framework:

- ipsas.php, the main entry to the application;
- application.spec, the application specification file;
- global/IpsasUser.php, the page user file;
- usermodule/LoginPage.php, the page class file
- usermodule/LoginPage.tpl, the page template file.
- usermodule/UserModule.php, the data module file.

The usermodule directory should be configured inaccessible to end-users because it contains sensitive application information.

Table 1 summarizes the entire Template Files with Class and Data Module Files for each module in PSAS.

Table 1: A Summary of Template Files with Class and Data Module Files for each module in PSAS.

Module	**Data Module Files (.php)**	**Template Files (.tpl)**	**Class Files (.php)**	**Description**
UserModule	UserModule.php	LoginPage.tpl NewPage.tpl EditPage.tpl	LoginPage.php NewPage.php EditPage.php	Logging in user. Creating new user Editing a user
AdminModule	AdminModule.php	AdminLayoutPage.tpl AdminMenuBar.tpl ViewAllPage.tpl ViewUserPage.tpl NewPage.tpl EditPage.tpl UpdateProfile.tpl ViewAllUserPage.tpl ExaminerAppPage.tpl NewSubmission.tpl EditSubmission.tpl AccessDenied.tpl ErrorPage.tpl	AdminLayoutPage.php AdminMenuBar.php ViewAllPage.php ViewUserPage.php NewPage.php EditPage.php UpdateProfile.php ViewAllUserPage.php ExaminerAppPage.php NewSubmission.php EditSubmission.php AccessDenied.php ErrorPage.php	
LecturerModule	LecturerModule.php	LecturerLayoutPage.tpl LecturerMenuBar.tpl UpdateProfile.tpl BorangA.tpl BorangB.tpl BorangC.tpl	LecturerLayoutPage.php LecturerMenuBar.php UpdateProfile.php BorangA.php BorangB.php BorangC.php	
StudentModule	StudentModule.php	StudentLayoutPage.tpl StudentMenuBar.tpl NewPage.tpl ViewAllPage.tpl NewStudentPage.tpl EditStudentPage.tpl EditStudentDetPage.tpl StudentModule.tpl	StudentLayoutPage.php StudentMenuBar.php NewPage.php ViewAllPage.php NewStudentPage.php EditStudentPage.php EditStudentDetPage.php StudentModule.php	
TitleModule	TitleModule.php	TitleLayoutPage.tpl TitleMenuBar.tpl NewPage.tpl EditPage.tpl ViewPage.tpl ByStatusPage.tpl	TitleLayoutPage.php TitleMenuBar.php NewPage.php EditPage.php ViewPage.php ByStatusPage.php	
ExaminerModule	ExaminerModule.php	ExaminerLayoutPage.tpl ExaminerMenuBar.tpl UpdateProfile.tpl MarkingPage.tpl ExaminationPage.tpl IExaminationPage.tpl IMarkingPage.tpl IExaminationListPage.tpl IMarkingEditPage.tpl	ExaminerLayoutPage.php ExaminerMenuBar.php UpdateProfile.php MarkingPage.php ExaminationPage.php IExaminationPage.php IMarkingPage.php IExaminationListPage.php IMarkingEditPage.php	

5. Conclusion

The web application framework was selected as the approach that will be used to develop the PSAS system mainly because it can speed up development process. Even though it takes a longer time to develop a system using an application framework at first, eventually when the framework has been used for several different systems development, the development time will be shortened. Furthermore, there is a lot of readily and freely available application framework which can be utilized to develop the system which makes the development process easier.

Increasing demands for web application has urged the e-business development organization to identify the best possible method to develop applications quickly without affecting the quality of the application in order to maintain the customers. Through this research project, it has been proven that robust and high quality web application development can be achieved through implementation of reuse based software engineering approach concentrating on web application framework. Implementation of this web application framework requires a high learning curve and training cost to learn the application. However, this implementation cost is a one time cost and organization can save a lot substantial amount of money through shorter software development time.

References:

Boehm B., *Software Engineering Economics* (Englewood Cliffs, NJ : Prentice-Hall, 1981

Sommerville I., *Software Engineering*, (7th Ed.). England: Pearson Education Limited, 2004

Wirfs-Brock R, and Johnson R. E.: Surveying Current Research in Object-Oriented Design. Commun. ACM 33(9): 104-124 (1990)

Package Aquarium, 2005, http://aquarium.sourceforge.net/api/public/aquarium-module.html#webapp Last access date: 5 September 2006.

Wildenauer J. C. php.MVC, http://phpmvc.net/ last access date: 5 September 2006.

Turing(2006), http://www.binarycloud.com/ last modified 17 April 2006, last access date: 5 September 2006.

ATK5, http://www.achievo.org/atk, last access date: 5 September 2006.

Smarty Template Engine (2006) http://smarty.php.net/ last access date: 5 September 2006.

PRADO Group (2006), http://www.pradosoft.com/ last access date: 5 September 2006.