RAPID APPLICATION DEVELOPMENT IN SYNERGY WITH PERSISTENCE FRAMEWORK

Choon How Choo and Sai Peck Lee
Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia

Keywords: Rapid application development, persistence framework, rapid prototyping, framework reuse.

Abstract: This paper proposes the concept, architecture, design and development of a rapid application development toolkit that will leverage on a persistence framework named PersistF, to subsequently provide an easy-to-use and customizable front-end web application development environment for software developers to perform rapid web application development. The proposed rapid application development toolkit consists of two main parts – RADEWeb and PersistF Configuration Wizard, to enable software developers not only to deliver their target web application within a shorter timeframe through an easy-to-use front-end environment, but also to achieve encapsulation of database access from the business objects of the web application.

1 INTRODUCTION

In fast changing environments, software developers must deliver software systems quickly to meet business delivery schedules, and not their own timetables. Spending months and years developing systems to high standards is fruitless if over time requirements change beyond recognition. Software development must serve its customers. Simple value-for-money systems that work are better than expensive and complex ones delivered late, over-budget, and that are difficult to maintain (Howard, 2002).

Rapid application development (RAD) has long been promised to be a boon to the computing community. The idea is to develop a method of designing software so that the whole process is quick, painless, and nearly effortless. The tools used should be easy to learn, powerful, and allow the designers to interface their freshly minted application with other applications, databases, and file types (Web Developer's Journal, 1997). While no universal definition of RAD exists, it can be characterized in two ways: as a methodology prescribing certain phases in software development (similar in principle to the spiral, iterative models of software construction), and as a class of tools that allow for speedy object development, graphical user interfaces, and reusable code for client/server applications (Agarwal et al., 2000).

Some companies offer products that provide some or all of the tools for RAD software development. These products include requirements gathering tools, prototyping tools, computer-aided software engineering tools, language development environments such as those for the Java platform, groupware for communication among development members, and testing tools. RAD usually embraces object-oriented programming methodology, which inherently fosters software reuse (Berry & Naumann, 2007). The most popular object-oriented programming languages, C++ and Java, are offered in visual programming packages often described as providing rapid application development.

What is a "persistence framework"? Before understanding the terminology, the two things that should be known are "Persistence" and "Framework". In computer science, "Persistence" refers to the characteristic of data that outlives the execution of the program that created it. Without this capability, data only exists in memory, and will be lost when the memory loses power, such as on computer shutdown. Persistence operations are usually separated into well-recognized categories, based on the types of data entries stored by the software. These categories are the ability to add (or create) new entries as well as view (or retrieve), edit (or update) and delete existing entries (Wikipedia, 2007a).

How Choo C. and Peck Lee S. (2008).
RAPID APPLICATION DEVELOPMENT IN SYNERGY WITH PERSISTENCE FRAMEWORK.
In *Proceedings of the Third International Conference on Software and Data Technologies - SE/GSDCA/MUSE*, pages 133-138
DOI: 10.5220/0001878301330138

A framework is a basic conceptual structure used to solve a complex issue. In the software context, a "framework" is a reusable design for a software system (or subsystem). A software framework may include support programs, code libraries, a scripting language, or other software to help develop and glue together the different components of a software project (Wikipedia, 2007b). Frameworks are also defined as a set of classes that embodies an abstract design for solutions to a family of related problems or a set of objects that collaborate to carry out a set of responsibilities for an application subsystem domain (Hyo et al., 2000). From the above explanation, it can be concluded that a persistence framework is one of the reusable designs.

A "persistence framework" moves the program data in its most natural form (in-memory objects) to and from a permanent data store (e.g. database) (RoseIndia, 2007). Most such frameworks require developers to maintain lots of meta-data describing how to map the object data into the relational database. However, with the advent of advanced language features (e.g. reflection), most of this meta-data can be obtained at runtime (Mertner, 2005).

2 MOTIVATION

Nowadays, most enterprise applications need access to a relational database (EL-Manzalawy, 2007). The Java standard for accessing relational databases is the JDBC APIs that utilize SQL as a data manipulation language. This approach of directly accessing a relational database from an object-oriented Java application was shown to be inefficient and introduces a problem called "object-relational impedance mismatch," or simply "impedance mismatch" for short, which refers to the differences between object-oriented technology and relational technology. An approach for solving the impedance mismatch problem is to introduce an abstract layer, called a persistence layer/framework, between the relational database and the object model of the application. This layer fully encapsulates the database access from the business objects (Ambler, 2006).

There are many persistence frameworks (both Open Source/Academia and Commercial) such as Hibernate (RoseIndia, 2007), JDO (RoseIndia, 2007), Castor (RoseIndia, 2007), and PersistF (Jusic & Lee, 2007). Most of the persistence frameworks require users (developers) to configure the framework's behaviour by providing enough information about the target application and the way in which its persistence aspects are to be handled in the context of a given framework. This decreases the overall productivity for the developer and complicates the development process by forcing the developer to learn the lingo of the target framework.

Tedious and error-prone configuration tasks and coding works may lead to costly, time consuming process, and increase time-to-market for the targeted application. PersistF, a framework prototype which was developed within our research programme, has been chosen as a basis for exploration, design and development of our proposed RAD toolkit due to the easy configuration of its framework behaviour. This research intends to develop a rapid application development toolkit that will leverage on PersistF to subsequently provide an easy-to-use and customizable front-end web application development environment for developers to perform rapid web application development.

3 RELATED WORK

Research work on the design and development of the framework diagram and framework-based RAD tool, named INTRAD (INTranet RAD) (Hyo et al., 2000) generates an application's source code based on the frameworks, analysis and design information. Therefore, programmers can develop applications easily and quickly by reusing not only source code but also design information. In Vidyadharan (2006), techniques for Rapid Application Development through the use of Data Binding and Object Persistence were discussed. He also outlined a general-purpose Data Binding framework that in combination with the choice of an appropriate persistence framework can facilitate Rapid Application Development. By using frameworks like Spring and Hibernate, Bolwidt and Partington (2006) produced a full working application in two days. They believed that the strong points benefited them most are good team development because of component separation. This work developed the RAD tool that produced a full working application in a very short time, but the biggest problem was the speed at which they could build the user interface (Bolwidt & Partington, 2006).

Bochicchio, Paiano and Paolini (1999) carried out a research called JWeb – a fast prototyping tool (Bochicchio et al., 1999). JWeb is an offspring of the Autoweb effort (Bochicchio et al., 1999). WARP (Web Application Rapid Prototyping) is a natural evolution of JWeb based on the new evolution of methodology and on fast-prototyping completely

online (Bochicchio & Fiore, 2004). Compared to existing tools, WARP is an aid for analysis and design of web applications because it introduces a new approach completely online (both the design and execution environment), so the author can immediately formulate and evaluate requirements, specification and designs.

In short, Hyo et al. (2000), Vidyadharan (2006), and Bolwidt & Partington (2006) showed the framework related approaches to achieve the features of Rapid Application Development. Autoweb, JWeb and WARP are series of web applications rapid prototyping tools, which are based upon the HDM model. These tools provide a fast development environment, an execution environment and the feature of fast-prototyping completely online.

4 ARCHITECTURE OF PROPOSED RAD TOOLKIT

In the context of this research work, software developers are programmers who want to develop web applications. They will be involved in two development phases using the proposed RAD toolkit – development through RADEWeb and development through Java IDE. RADEWeb is a front-end web application development environment of the toolkit which has to be installed in a server to enable web application development and rapid prototyping through the web. Java IDE is a Java integrated development environment that provides comprehensive facilities to software developers for application development which is usually installed in the developer's workstation. Figure 1 together with Figure 2 shows the overview of the proposed RAD toolkit's concept and architecture which aim to perform rapid web application development in synergy with the persistence framework, PersistF.

In the phase of development through RADEWeb, developers can access to RADEWeb through their web browsers. A web application workspace is allocated for each web application to be developed. Functions of metadata manipulation (add, edit, view and delete), rapid prototyping, web application code generation, and project archive downloading are available in this workspace.

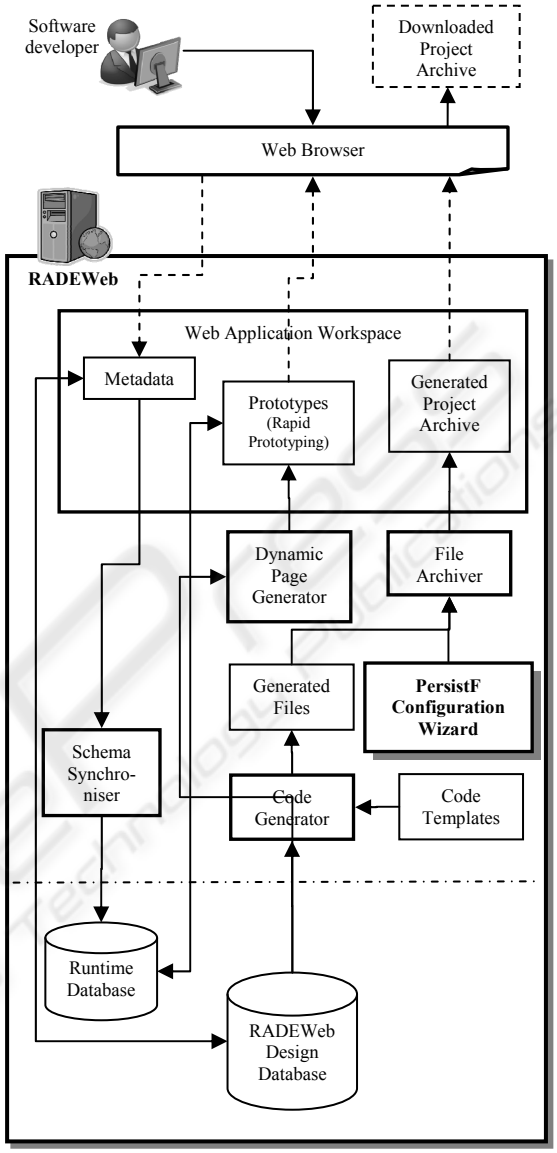

Figure 1: Overview of the proposed RAD toolkit for development through RADEWeb.

The four main components in RADEWeb are Schema Synchroniser, Dynamic Page Generator, Code Generator, File Archiver. Schema Synchroniser plays an important role in synchronising the schema of the runtime database based on the schema that has been captured in the metadata. For the purpose of rapid prototyping, Dynamic Page Generator generates the requested prototypes based on the design information of the web application stored in RADEWeb's design database. Code Generator generates the Tomcat-based web application's skeleton source code files, builing upon the persistence framework, PersistF, by

referring to the design information described in the metadata kept in the RADEWeb design database together with the code templates. File Archiver combines the generated files with PersistF configuration wizard of the web application into a project archive file for easier transportation from RADEWeb to the developer's workstation.

In the phase of development through Java IDE (Figure 2), further web application development will be carried out in the developer's workstation.

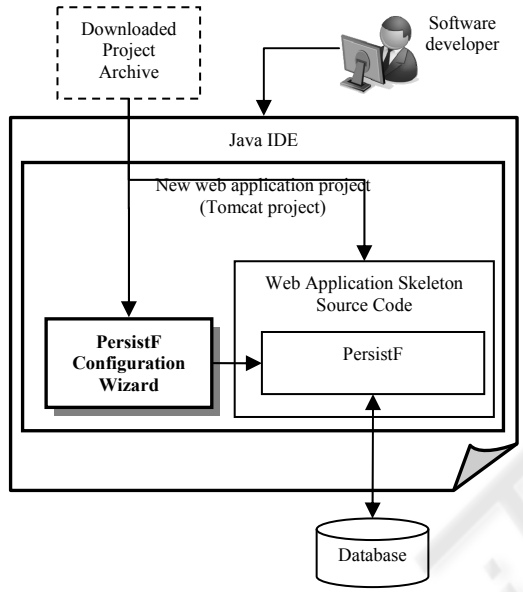

Figure 2: Overview of the proposed RAD toolkit for development through Java IDE in synergy with PersistF.

By using a Java IDE such as Eclipse or NetBeans, he/she can import project archive into a new web application project – Tomcat project, to proceed with further development as needed. The developer can configure PersistF's behaviour by using the PersistF configuration wizard (which has previously been archived as a jar file) along with the web application's source code in the project folder. Configuration through the wizard avoids tedious and error-prone manual coding task which may lead to costly and time consuming process. During the web application development, PersistF plays an important role in taking care of all the persistence-related work on behalf of the developer (Jusic & Lee, 2007). This saves a lot of time from writing hand-coding SQL and the supporting code to turn it into a Java object. The resulting web application's skeleton source code and the encapsulated database access layer (PersistF) contribute to providing RAD features in subsequent web application development.

5 DESIGN & IMPLEMENTATION

The aim of this section is to present the design and implementation of a rapid application development toolkit. One of the ways to increase productivity is to reduce complexity (Arthur, 2006). With the simple development process as shown in Figure 3, together with the code generation facility and configuration wizard, it will help to reduce coding errors during development and deployment of web applications. In addition, it will also aid developers in building web applications with a set of services without writing code.

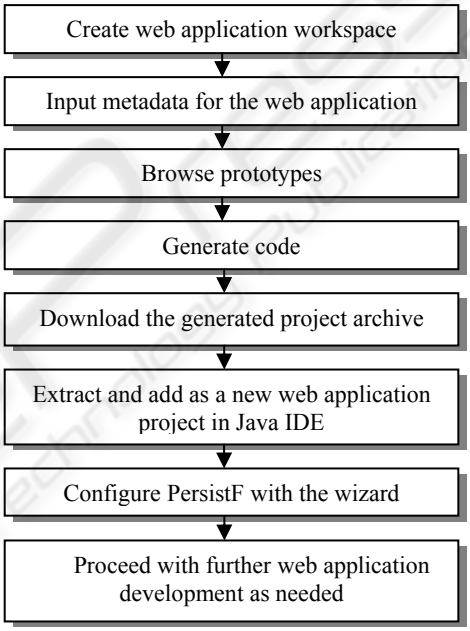

Figure 3: Process of the web application development with the proposed RAD toolkit.

5.1 RADEWeb

RADEWeb, a front-end web application development environment that allows the software developer to create a workspace for the web application to be developed, input metadata for the web application, browse web application prototypes, generate the web application's source code based on the captured metadata and reusable design infrastructure of PersistF, as well as download the web application's generated code compressed as a .zip archive. RADEWeb enables multi users to access and develop web applications anywhere and anytime without spending time with tedious development environment configurations.

5.1.1 Web Application Workspace

A software developer is allowed to create more than one web application workspace. When a new web application workspace is to be created, the name of the web application needs to be entered, and will be automatically defaulted as the name for its database schema. Every successfully created web application workspace will have its own runtime database schema for the purpose of rapid prototyping.

5.1.2 Metadata

The metadata for the web application may include, but not limited to, class names for the web application, attribute names for each class, data type and input type for each attribute (for interface customisation). All of these metadata will be stored in the RADEWeb design database.

5.1.3 Rapid Prototyping

From the metadata of the web application, software developers can browse the web application prototypes which were generated dynamically based on the captured metadata. They can test and feel the freshly minted web application instantly and edit the metadata as needed.

5.1.4 Code Generation

Once the software developer is satisfied with the prototypes, he/she can generate skeleton code of the web appication based on the captured metadata in synergy with PersistF. This saves developers from spending time on repetitive and error-prone manual coding work which may lead to costly, time consuming process, and increase time-to-market of the target web application. The generated skeleton code has an architecture that indirectly reuses PersistF defined design patterns, and thus, reducing the risk of coding manually.

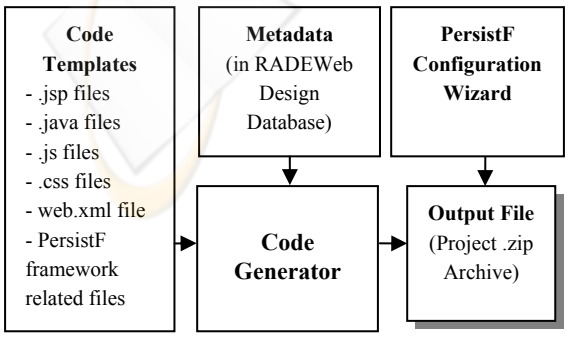

Figure 4: Overview of Code Generation.

Figure 4 showed an overview of code generation. There are two inputs for the code generator - metadata which is stored in the RADEWeb design database; and code templates. Only metadata and code templates which are related to the specific web application will be retrieved and analysed for the purpose of code generation.

5.1.5 Downloadable Project .zip Archive

Generated source code of a web application will be compressed as a .zip archive so that the developer can download it easily from RADEWeb. This compressed archive contains the web application's skeleton source code which can be extracted and added as a new project in Java IDE such as Eclipse or NetBean for further development as needed.

5.2 PersistF Configuration Wizard

PersistF configuration wizard forms part of the proposed RAD toolkit. It is used to configure the settings of PersistF for the web application to be generated. By using this wizard, developers only have to follow the instructions; choose an appropriate setting and edit it as needed from the GUI interface instead of having to edit code manually such as using a traditional text editor to configure the framework's runtime engine with sufficient information about the domain application. This makes error-prone configuration tasks much easier with the wizard. It also prevents developers from spending time on fixing setting errors that frequently occurs during configuration.

6 CONCLUSIONS

This paper proposed a rapid application development toolkit for developing web applications, applying a persistence framework that fully encapsulates the database access from the business objects. The simple development process defined has provided a guide to developers in performing rapid web application development with the toolkit. The research has benefited from some RAD approaches proposed in previous works. Besides allowing rapid web application development, it is also the most effective way for software developers to avoid having to perform error-prone manual coding tasks during web application development. Future works could be on enhancing this toolkit to support development of complex web applications.

REFERENCES

Agarwal, R., Prasad, J., Tanniru M., Lynch, J. 2000, 'Risks of Rapid Application Development', *Communications of the ACM*. Association for Computing Machinery, New York, USA, vol.43(11), pp 177-188. Retrieved July 20, 2007, from ACM (Association for Computing Machinery) Digital Library Online Database

Ambler, S.W. 2006, *Encapsulating Database Access: An Agile "Best" Practice*. Retrieved July 25, 2007, from: http://www.agiledata.org/essays/implementationStrategies.html

Arthur, J. & Azadegan, S. 2006, 'Spring Framework for rapid open source J2EE Web Application Development - A case study', Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005 and First ACIS International Workshop on Self-Assembling Wireless Networks. SNPD/SAWN 2005. Sixth International Conference. IEEE, Sci. Applications Int. Corp., San Diego, CA, USA, pp. 90-95. Retrieved December 14, 2007, from IEEE Xplore™ Online Database

Berry, V. & Naumann, A. 2007, *What is rapid application development - a definition from Whatis_com.* Retrieved July 27, 2007, from: http://searchsoftwarequality.techtarget.com/sDefinition/0,,sid92_gci214246,00.html

Bochicchio, M., Paiano, R., Paolini, P. 1999, 'JWeb: an HDM environment for fast development of web applications', *Multimedia Computing and Systems, 1999. IEEE International Conference*. Florence, vol. 2, pp. 809-813. Retrieved December 12, 2007, from IEEE Xplore™ Online Database

Bochicchio, M., & Fiore, N. 2004, 'WARP: Web Application Rapid Prototyping', *Proceedings of the 2004 ACM symposium on Applied computing*. Association for Computing Machinery, Nicosia, Cyprus, pp. 1670-1676. Retrieved December 13, 2007, from ACM (Association for Computing Machinery) Digital Library Online Database

Bochicchio, M., & Fiore, N. 2005, 'WARP for Re-Engineering of Web Applications', *Proceedings of the sixteenth ACM conference on Hypertext and hypermedia*. Association for Computing Machinery, Salzburg, Austria, pp. 295-297. Retrieved December 14, 2007, from ACM (Association for Computing Machinery) Digital Library Online Database

Bolwidt, E. & Partington, V. 2006, *Java with Spring just as productive as a 4GL RAD tool*. Retrieved July 20, 2007, from http://www.xebia.com/file_db/File/artikel%20Erwin%20Bolwidt%20en%20Vincent%20Partington.pdf

Corcho, O., López-Cima, A., Gómez-Pérez, A. 2006, 'The ODESeW 2.0 Semantic Web application framework', *Proceedings of the 15th international conference on World Wide Web*, Association for Computing Machinery, Edinburgh, Scotland, pp. 1049-1050. Retrieved February 14, 2007, from ACM (Association for Computing Machinery) Digital Library Online Database

EL-Manzalawy, Y. 2007, *Accessing Data Through Persistence Frameworks*. Retrieved July 27, 2007, from http://www.agiledata.org/essays/implementationStrategies.html

Howard A. 2002, 'Rapid Application Development: rough and dirty or value-for-money engineering?', Communications of the ACM., vol. 45, issue 10, pp. 27-29. Retrieved December 13, 2007, from ACM (Association for Computing Machinery) Digital Library Online Database

Hyo, T.J., Dong, K.K., Young, J.Y., Lee, J.Y. 2000, 'A design and implementation of object-oriented framework-based RAD tool (INTRAD)', *Systems, Man, and Cybernetics, 2000 IEEE International Conference.*. Nashville, TN, vol. 3, pp. 2057-2061. Retrieved July 31, 2007, from IEEE Xplore™ Online Database

Jusic, S. & Lee, S.P. 2007, 'PersistF: A Transparent Persistence Framework with Architecture Applying Design Patterns', *Issues in Informing Science and Information Technology*, Informing Science Institute, Ljubljana, Slovenia, vol. 3, pp. 767-779. Retrieved July 26, 2007, from http://proceedings.informingscience.org/InSITE2007/IISITv4p767-779Jusi281.pdf

Mertner, M. 2005, *What is a Persistence Framework*, Retrieved July 25, 2007, from http://www.mertner.com/confluence/display/Gentle/1+-+What+is+a+ Persistence+Framework

RoseIndia. 2007, *What is Persistence Framework?* Retrieved July 25, 2007, from http://www.roseindia.net/enterprise/persistenceframework.shtml

Vidyadharan, R. 2006, *Rapid Application Development With Data Binding and Object Persistence*. Retrieved July 20, 2007, from http://www.sptci.com/products/articles/rad.pdf

Web Developer's Journal. 1997, *Rapid Application Development*. Retrieved July 26, 2007, from http://www.webdevelopersjournal.com/articles/rad.htm

Wikipedia. 2007a, *Persistence (computer science)*. Retrieved July 25, 2007, from http://en.wikipedia.org/wiki/Persistence_%28computer_science%29

Wikipedia. 2007b, *Software framework*. Retrieved July 26, 2007, from http://en.wikipedia.org/wiki/Software_framework