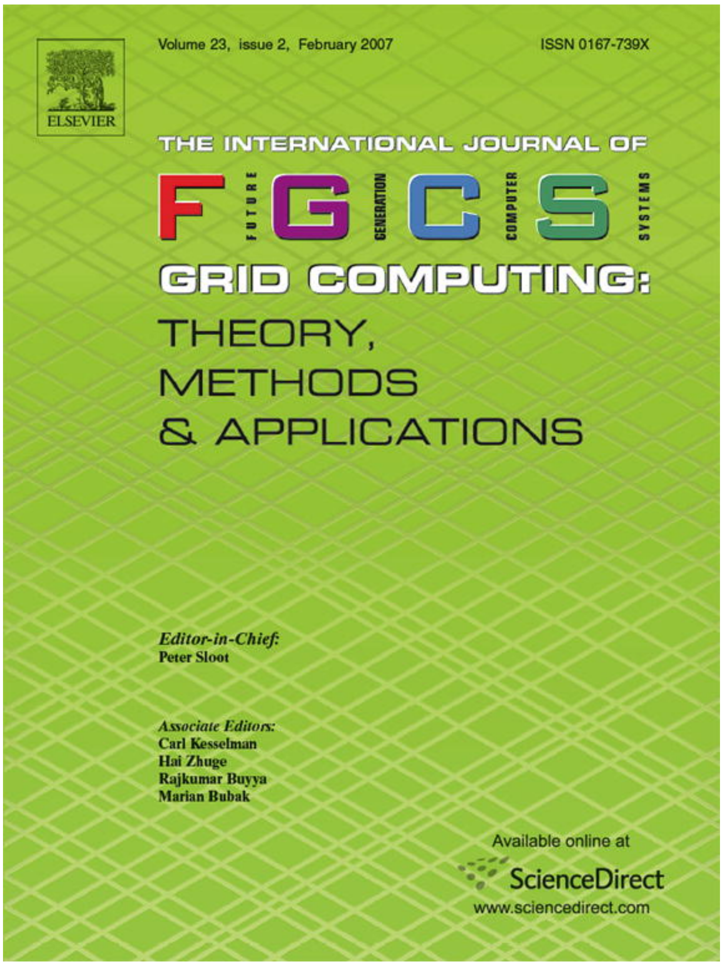

Volume 23, issue 2, February 2007
ISSN 0167-739X
ELSEVIER
THE INTERNATIONAL JOURNAL OF
F FUTURE G GENERATION C COMPUTER S SYSTEMS
GRID COMPUTING:
THEORY,
METHODS
& APPLICATIONS
Editor-in-Chief:
Peter Sloot
Associate Editors:
Carl Kesselman
Hai Zhuge
Rajkumar Buyya
Marian Bubak
Available online at
ScienceDirect
www.sciencedirect.com

Available online at www.sciencedirect.com
ScienceDirect
Future Generation Computer Systems 23 (2007) 263–268
FGCS
FUTURE GENERATION COMPUTER SYSTEMS
www.elsevier.com/locate/fgcs

Web framework with Java and XML in multi-tiers for productivity

Gun Ho Lee*, Junsu Jung

Department of Industrial and Information Systems, Soongsil University, Sangdo-5dong, Dongjak-Gu, Seoul 156-743, Republic of Korea

Received 28 February 2006; received in revised form 14 April 2006; accepted 16 May 2006
Available online 7 August 2006

Abstract

As web-based business systems get larger and more complex, an efficient development framework is an important issue for developers. An efficient web framework improves the quality of multi-tier systems and productivity in the development in distributed computing environments. We provide a framework approach with Java for 3-tiers (Client/Web, Business/EIS), which allows transporting, transforming, or integrating data through the systems using structured-information based on textual format of XML. In the framework for 3-tiers system development with Java, XML is used as a mediator for application data communication as well as a descriptor for configuration. The framework in this paper simplifies workflow and reduces time and effort to develop the systems. We analyze and compare workflows of the framework and productivities in terms of lines of code and effort (man-months) needed to develop a system with identical functions.

Keywords: Web framework; Multi-tiers; Java; XML; Productivity

1. Introduction

As web based systems get larger and more complex in a multi-tier environment, it takes a lot of time and effort to develop the multi-tier system. An efficient framework approach for the multi tier system, which includes client tier, web tier, business tier, and Enterprise Information System (EIS) tier, is an important issue for developers in distributed computing environments [1,5,9,12].

Framework is a promising technology for reifying proven software designs and implementations in order to reduce the cost and time for multi-tier system development and to improve the quality of the system [13–15,26]. The framework is a key component for achieving reuse of analysis, design, and implementation [21]. A framework is a reusable, semi-completed application that can be specialized to produce custom applications [6,7]. The primary benefits of frameworks stem from modularity, reusability, extensibility, and inversion of control. In contrast to earlier object-oriented reuse techniques based on class libraries, frameworks are targeted for particular business units such as data processing or cellular communications, and application domains such as user interfaces or real time avionics [8,16].

Recently, solutions for the tiers have been studied and developed such as Velocity [30], Struts [3], Spring [22], WebWork [29], and Tapestry [10]. The above frameworks have some weaknesses:

– Since precedence constraints exist between the development of tiers with the above frameworks, the tiers should be developed sequentially.
– Adaptability and extensibility are limited in improving functions of the overall system.
– Developers cannot systematically verify and validate functions between the tiers but can with personal skill or with intuition for each application. The roles of tiers are responsible for each application.

Hence, the framework should be improved as follows:

– Architectures of tiers should technically be independent of each other for adaptability and extensibility.
– Tiers should not have any dependency relation in development process in order to develop the tiers in parallel, e.g., layout and design of window and business logic should be developed for each tier in parallel.
– Interface and communication between tiers should be easy even for a large size of tiers.

* Corresponding author. Tel.: +82 2 820 0697.
E-mail address: ghlee@ssu.ac.kr (G.H. Lee).

0167-739X/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2006.05.010

– The design, layout, and control of the user interface should be separated and independent of each other since a developer frequently modifies the user interface in a presentation layer.
– An object to control a variety of resources such as connection, session, configuration, etc. should improve efficiency in the development of tiers.

Companies attempting to build or use large scale reusable frameworks often fail unless they recognize and resolve challenges such as development effort, learning curve, integrity, maintainability, validation and defect removal, efficiency, and lack of standards [8]. However, existing application frameworks require considerable investment of effort to learn. An application is developed based on the integration of multiple frameworks together with class libraries, legacy systems, and existing components.

The user interface, connecting middleware, and business logic and data are tightly coupled through hard-coded interfaces although the 3-tier client/server model improves the scalability and manageability of distributed computing. XML has a self-documenting format that describes structure and field names as well as specific values and is platform-independent, thus relatively immune to changes in technology [24,25,27]. The universal format of data on web documents is simple to understand and allows developers to describe, deliver, and integrate rich and structured data from any application in a standard and consistent way [20].

The web frameworks of Struts, Spring, WebWork, and Tapestry are designed based on eXtensible Markup Language (XML) focusing on conceptual modeling of component-based framework in 3 tiers. However, the frameworks use XML as a descriptor to define a configuration that constitutes a framework but does not use XML for the data communication between the tiers.

Struts [3] has strengths such as overriding of a core class by open architecture, general use of Taglib in other applications, many references and user groups, etc. However, it is inconsistent in a policy to transport parameters through tiers and abnormal validation, debugging, and testing by separating action and actionForm. Action is dependent on a servlet container, which provides a variety of Taglib but cannot reuse standard HTML.

Spring [22] supports a POJO (plain old Java object)-based model for an easy use of J2EE [11], business interface not related to a subroutine like POJO or local EJB, and overriding of a core class by open architecture. Spring is also inconsistent in a policy to transport parameter through tiers. Action is dependent on servlet container.

WebWork [29] is not efficient for validating, and requires a lot of time and cost for developers to learn and understand the framework.

With the existing frameworks, a developer has endeavored to transform data into XML, validation, interfaces between tiers, etc. in person [28]. We design the framework in order that tiers can communicate with each other by the medium of XML. Thus, the framework takes advantage of XML features such as flexibility, extensibility, and platform independence. We develop a framework model, which is called ‘ezFrame’, using XML in J2EE multi-tier of Sun Inc. [23] based on a model of Model-View-Control (MVC).

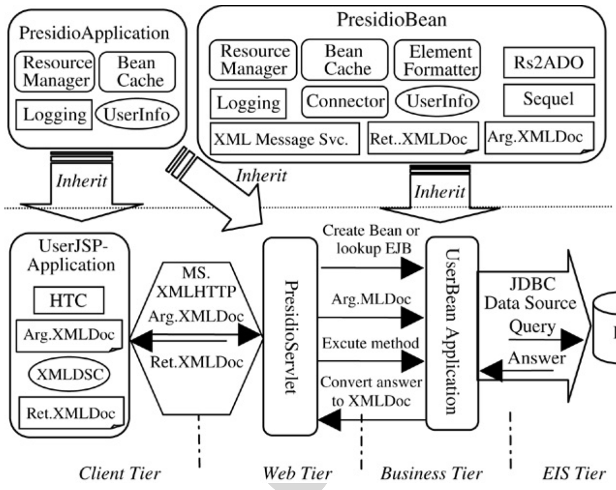


Fig. 1. ezFrame architecture.

2. ezFrame

In ezFrame, we use XML not only for descriptor files to define configuration but also as a tool for data communication between tiers. In data communication between tiers, we make the complex structure of the framework layer simple and easy for developers to understand. We reduce the size of the framework by representing user request data and its response data into XML, which furthermore makes it easy to interface frameworks, corresponding tiers and improve the framework flexibility. Fig. 1 represents the ezFrame architecture.

For MS Internet Explorer web browser users, the ezFrame uses the technologies of MSXML, Data Source Object (DSO) [19], and data binding [18] for a presentation. It is also possible to apply the approach in this paper to the technology of Taglib, Applet [31], Application, XSLT (Java API for XML Processing), JAXB (Java Architecture for XML Binding), and Velocity [30].

2.1. The ezFrame elements

The ezFrame includes 17 major elements:

PresidioApplication is a super servlet. Thus every application JSP and servlet inherits from PresidioApplication. It defines methods for a common use of session with method of utility including preference setups like login page, main page, and connecting DB, definition of Bean or EJB (Enterprise Java Beans), logging, etc.

PresidioBean as a standard type of Bean or EJB inherits from Bean or EJB of every application. PresidioBean includes the various definitions such as a common use of session, user preference, various utility methods, sub-Bean or sub-EJB, logging, control of database connection, SQL loading, transformation of result data into XML, messages, etc.

Arg. XML Doc is an XML document containing user request information, which includes user action to be transferred from client tier to web or business tiers.

Ret. XML Doc is an XML document containing request output which includes message, ResultSet, etc. to be transferred from web and business tier to client tier.

PresidioServlet is a servlet, which defines a skeleton of a standard data flow. It performs Bean or EJB according to the requests based on Arg. XML Doc containing parameters given by the client and returns Ret. XML Doc as an output to the client.

UserJSPApplication defines a basic page frame inherited from PresidioApplication of a super-servlet. Since the JSP makes data binding by XML DSO [11,19], the data and the composition of controls are processed separately. JSP also defines XML DSO internally. The controls, which are made up with HTML elements by mapping the XML DSO defined, use various types of HTCs [17] and TagLib embedded in JSP.

UserBeanApplication, which is a kind of Bean or EJB in the business tier, contains a method of business logic that should be defined by developers. It makes a basic frame by inheritance from PresidoBean.

The method for business logic determines an XML file including SQL to perform. In Arg. XML Doc received parameters, the method has a name type of "excute" + [*Action name*]. The method is automatically called and performed according to an action in PresidioServlet. Bean or EJB to process common logic or sublogic are made up separately. The Bean or EJB transform connection, user information, etc. to XML Doc or other type of parameters for processing.

ResourceManager is a class, which controls message of properties or run time configuration, and loads a variety of information according to a type of virtual machine action based on a defined mechanism.

Logging is a logging utility employing Log4J [4] which determines logging levels according to the importance, i.e., information, debug, warning, error, and fatal error, and also according to application sizes, i.e., system application, group application, and individual application.

UserInfo is an object that is generated with the user information and authorization information after a user is authorized. The object is kept and managed in a session and can be transformed to XML elements and be sent to each tier if needed.

Bean Caching. Once a bean instance is generated initially by a cache algorithm, the instance is cached, returned, and reused for following requests.

Connector is a class for DB connection that makes, controls, and maintains DB connections and performs query and transaction control. Special types of connections can be used in addition to the basic types of DataSource, Java DB connection, EJB connection, XML DSO, IBM host call, etc. It is intended for a user not to use the low level API for accesses to the DB but to do everything related to the DB connection like creating ResultSet, statement, and searching current counts of data reference.

Sequel is a management utility class of SQL resource, which controls SQL sentence coded as a form of 'key:value' in external XML file and processes parameters, multiple SQL, block substitution, etc.

Rs2ADO is a utility class transforming ResultSet to RecordSet elements for employing XML DSO and can print out RecordSet, metadata, and template.

XML DSO supports XML data source by XML data islands in web browser versions more than or equal to Internet Explorer 5. It defines embedded XML data with <xml></xml> elements in HTML pages and also references the external XML data source by using the src property, which does data binding to MSHTML.

XML Message Svc. generates messages such as information, warning, error, fatal error, exception, and trace. The generated message is an XML element in Arg. XML Doc or Ret. XML Doc and is used as a basis of decision making for communication.

ElementFormatter defines an output form of a result set of personnel information and is used by Rs2ADO. Each sub-tag of row-tag means column label of ResultSet queried. Index and flag tag as an additional information show the property of each row.

2.2. *Framework action flow*

When a user requests any information, the information is generated as an XML document called Arg. XML Doc For example, when the user searches personnel information by entering an identification number for authorization to the personnel information system, Arg. XML Doc includes the user's name, the identification number searched, object type, name of target object, and method name.

Arg. XML Doc is transferred to PresidioServlet which is defined as a standard data flow by XMLHTTP [20]. Client passes a stream of Arg. XML Doc to PresidioServlet and then PresidioServlet renews a session of user information in the Arg. XML Doc Then the remote system checks and forwards the request to process it at the next step or the remote system may create a target object based on Arg. XML Doc and return its parameter to Arg. XML doc.

Then a specified method is called and the method collects information from Arg. XML Doc and produces the final Ret. XML Doc by creating connection, performing SQL, etc. Ret. XML Doc includes a ResultSet about the personnel information searched and the message to send to the client.

PresidioServlet deletes UserBeanApplication, which is an object created, after performing the method in the UserBeanApplication to obtain Ret. XML Doc and then transport the Ret. XML Doc to the UserJSPApplication in the client tier via XMLHTTP. The UserJSPApplication properly reorganizes data in the Ret. XML Doc for readability after verification. The reorganized data in Ret. XML Doc is used as XML DSO and shown upon a screen after data binding.

Fig. 2 represents action flow of ezFrame. JavaScript collects parameters, information of target action mapping, etc. in the client tier into Arg. XML Doc for MSXML API. PresidioServlet transports the Arg. XML Doc to a business object that processes the information in the Arg. XML Doc in accordance with the logic. Java API collects an output for the query and other information into Ret. XML Doc that is

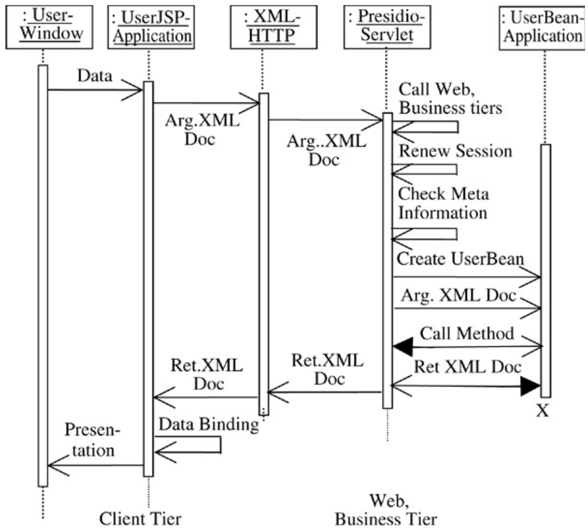


Fig. 2. Action flow of ezFrame.

returned to the client tier via PresidioServlet. The Ret. XML Doc is finally obtained in the client tier.

3. Comparison with previous work

Although existing frameworks are designed to make a use of XML, the XML is used only for descriptor files to define configuration of the framework but not for a medium of communication between tiers. The existing frameworks are too complex to interact between tiers, which in turn requires and controls a number of descriptors for XML. It takes a lot of time and effort to learn the complex architecture of the frameworks.

In this paper, we maximize the flexibility of a tier to interface a framework with the others of a different vendor by representing request data and response data with XML and integrating the complex structure of frameworks in each tier into an object to control XML. We use DSO, data binding, etc. which is specialized in MS Internet Explorer.

3.1. Workflow of framework

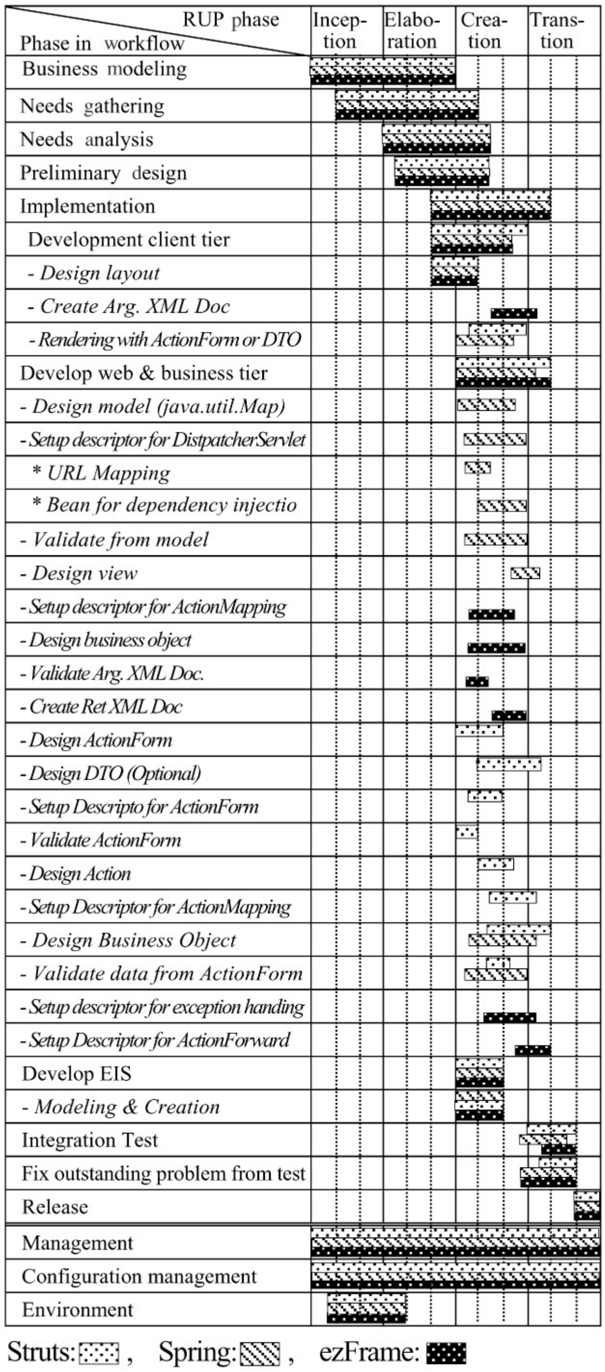
Table 1 shows a Gantt chart of workflows of three frameworks, which vertically categorizes implementation processes of each framework, i.e., client, web, business, and EIS tiers, and horizontally shows Rational Unified Process (RUP) of each framework. The development workflows over RUP in ezFrame are shorter and simpler than those of the two other frameworks.

3.2. Development efforts of framework

In this study, tier development using ezFrame and test processes may be performed in parallel since tiers are free from precedence constraints. Hence, the period of the development time is shortened.

We compare development efforts of four types of windows with each framework by COCOMO [2] in Table 2. Although Struts reduces the size of the client part by using Taglib, it takes a lot of effort to develop the ActionForm for a window and the

Table 1
Workflow comparison for frameworks



Data Transport Object (DTO). Spring improves the server tier but just employs Struts for the client tier. Table 2 shows LOC, FSP, and MM in developments of list browse, single browse, multi-entry, and master/slave entry by three frameworks. List browse is a window for searching a list of recodes. Single entry is a window that inputs, modifies, and deletes data for a single record. Multi-entry is a window for multiple records. Master/slave entry is a window which searches and modifies a number of slave type records for a master type record.

Table 2
Development efforts

Framework	Unit	List browse		Single browse		Multi-entry		Master/slave entry	
		Client	W/B	Client	W/B	Client	W/B	Client	W/B
Struts	LOC	278	668	1022	1360	1112	1762	1984	2399
	MM	0.63	1.57	2.46	3.31	2.68	4.35	4.93	6.02
	FSP	0.30	0.53	0.70	0.84	0.74	1.0	1.08	1.22
Spring	LOC	278	372	1022	1108	1112	1550	1984	1944
	MM	0.66	0.85	2.46	2.68	2.68	3.80	4.93	4.82
	FSP	0.30	0.36	0.70	0.74	0.74	0.92	1.08	1.06
ezFrame	LOC	303	274	882	712	943	985	1685	1216
	MM	0.69	0.62	2.10	1.68	2.26	2.36	4.15	2.95
	FSP	0.32	0.29	0.63	0.55	0.66	0.68	0.97	0.78

W/B: Web/Business, LOC: Line Of Code, MM: Man-Month, FSP: Full Software Personnel.

4. Case study

We apply ezFrame in developing four web-based applications and count LOC, Work Hour(WH), and FSP of each project as shown in Table 3.

(1) Oil product trade management system. This system is a decision support system which provides the trade information such as trade volume, recent prices, the product volumes to sell or buy, etc. for the trader to determine oil product prices. Two Java programmers and one designer developed the system after one week of education and training on ezFrame. It took 16 work hours to develop a window for entry/modification and 4 work hours to do a window for search, respectively.
(2) Supply chain management system for fashion clothes. This system includes subsystems for management such as production, marketing, order, shipping, handling, subcontractor, customer community, shop, etc. We applied ezFrame only to develop the subsystem of customer community management. It took six weeks for one Java programmer to develop the customer community management.
(3) Management information system of still company. This system includes subsystems of managements for import and export, global marketing, customer relational, etc. We educated and trained one designer and two Java programmers on ezFrame for a week. After the training and education, it took 4 work hours to complete a window unit for a search and 2 work days to complete an entry and modification window, respectively.
(4) Web-based management information system.

We integrated various applications of each branch into a management information system for an integrated management model. The system includes global risk management, customer service management, marketing management, finance and accounting management, etc. One designer and two Java programmers were trained on DBA and ezFrame. After two weeks' experience of ezFrame, they complete 130 window units in 10 weeks.

Table 3 represents LOC/FSP in each project. Although the ratios of LOC/FSP are diverse due to the project complexity, the developer's domain knowledge, the duration of education and training, etc., productivity of the developers is doubled. The development productivity is not dependent on developers' skill but capability of the framework use for related parts.

Table 3
LOC/FSP for a development

Project #	Unit	Average in master/slave			
		Client	LOC/FSP	W/B	LOC/FSP
1	LOC	4422	2440.04	[a]	–
	WH	11.43		[a]	
	FSP	1.812		[a]	
2	LOC	5477	2629.38	2714	2056.06
	WH	14.31		6.85	
	FSP	2.083		1.32	
3	LOC	1226	1551.90	2252	1924.79
	WH	2.97		5.63	
	FSP	0.79		1.17	
4	LOC	2626	1959.70	395	1039.47
	WH	6.61		0.90	
	FSP	1.34		0.38	
5	LOC	1265	1581.25	1853	1816.67
	WH	3.07		4.59	
	FSP	0.80		1.02	
6	LOC	3793	2312.80	1922	1830.48
	WH	9.73		4.77	
	FSP	1.64		1.05	

WH: Work Hours.
[a] Call from IBM host.

5. Summary and conclusion

The ezFrame framework described in this paper uses XML for data communication as well as a descriptor file to set up configuration for each part of the framework. We simplified and reduced the framework by introducing an integrated XML management object instead of representing request data and its response data as XML, which in turn makes the tier structure of the framework more complex.

We have tangible and intangible advantages in ezFrame. In ezFrame, architectures of tiers are technically independent so that the tiers can be easily adapted and extended. Developers can work parallelly for each tier through the development processes since the design, layout, control of the user interface, etc. are separated and independent of each other. Case studies show that developers can use ezFrame after a short time of learning and training. The developers who use ezFrame also complete the application development with less effort and in shorter periods than those who use the existing web frameworks.

In addition to strengths of framework, stability, reusability, shorter lines of codes, reduced debugging time and efforts, we can take advantages of XML-based structures like flexibility, efficiency of information transporting, extensibility, etc. in ezFrame.

Acknowledgement

This research is supported by Soong Sil University.

References

[1] E. Laure, OpusJava: A Java framework for distributed high performance computing, Future Generation Computer Systems 18 (2) (2001) 235–251.
[2] B. Boehm, Software Engineering Economics, 1981.
[3] C.R. McClanahan, Apache Struts project, http://struts.apache.org.
[4] C. Gülcü, M. Steiner, Apache Log4J, http://logging.apache.org/log4j/docs.
[5] D. Brugali, K. Sycara, Frameworks and pattern languages: an intriguing relationship, ACM Computing Surveys (March) (2000).
[6] D. Baumer, G. Gryczan, R. Knoll, C. Lilienthal, D. Riehle, H. Zullighoven, Framework development for large systems, Communications of the ACM (October) (1997).
[7] E. Ebner, W. Shao, W.-T. Tsai, The five-module framework for Internet application development, ACM Computing Surveys (March) (2000).
[8] M.E. Fayad, D.C. Schmidit, R.E. Johnson, Object-Oriented Application Frameworks: Problems and Perspectives, Wiley, New York, 1997.
[9] H. Mili, H. Sahraoui, I. Benyahia, Representing and querying reusable object frameworks, in: Proceedings of the 1997 Symposium on Software Reusability.
[10] Jakarta Tapestry, http://jakarta.apache.org/tapestry.
[11] Java 2 Platform, Enterprise Edition (J2EE), http://java.sun.com/j2ee/index.jsp.
[12] T. Haupt, E. Akarsu, G. Fox, WebFlow: A framework for web based metacomputing, Future Generation Computer Systems 16 (5) (2000) 445–451.
[13] J. Hancock, Application frameworks before system frameworks, in: 2000 Proceedings of the Conference on Object-Oriented Programming, Systems, Languages, and Applications (Addendum).
[14] J. Sholberg, J. Illback, Application integration framework for a large business process redesign project, in: 2000 Proceedings of the Conference on Object-Oriented Programming, Systems, Languages, and Applications (Addendum).
[15] M.E. Markiewicz, C.J.P. de Lucena, Object oriented framework development, in: Crossroads, Summer 2001.
[16] M. Fayad, D.C. Schmidt, Object-oriented application frameworks, Communications of the ACM (October) (1997).
[17] MS's Behaviors, http://msdn.microsoft.com/library/default.asp?url=/workshop/author/behaviors/behaviors_node_entry.asp.
[18] MS's Data Binding Architecture, http://msdn.microsoft.com/library/default.asp?url=/workshop/author/databind/architecture.asp.
[19] MS's Data Source Object, http://msdn.microsoft.Com/library/default.asp?url=/workshop/author/databind/datasources.asp.
[20] MS's XML HTTP, http://msdn.microsoft.com/library/default.asp?url=/library/enus/wceappservices5/html/wce50oriXMLHTTP.asp.
[21] R.E. Johnson, Frameworks = (components + patterns), Communications of the ACM (October) (1997).
[22] Spring Framework, http://www.springframework.org.
[23] SUN's Distributed Multi-tiered Applications, http://java.sun.com/j2ee/1.4/docs/tutorial/doc/Overview2.html.
[24] SUN's Java Architecture for XML Binding (JAXB), http://java.sun.com/developer/technicalArticles/WebServices/jaxb/index.html.
[25] SUN's XML Solutions with JavaServer[tm] Pages, http://java.sun.com/products/jsp/html/JSPXML.html.
[26] T. Hansen, Development of successful object-oriented frameworks, in: 1997 ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (Addendum).
[27] V. Turau, A framework for automatic generation of web-based data entry applications based on XML. in: Symposium on Applied Computing, Proceedings of the 2002 ACM symposium on Applied computing.
[28] Web frameworks comparison, https://equinox.dev.java.net/framework-comparison.
[29] WebWork, http://www.opensymphony.com/webwork.
[30] Velocity, http://jakarta.apache.org/velocity.
[31] Applets, http://java.sun.com/applets.

Gun Ho Lee is Associate Professor of Industrial and Information Systems Engineering at Soongsil University, Seoul, South Korea. He obtained a Ph.D. degree in Industrial Engineering from the University of Iowa, Iowa City, Iowa, USA. Dr. Lee has worked and consulted for a number of corporations and consortia. His research interests include object oriented development, framework, software development process, web application test for security, data mining, etc. His papers have appeared in software engineering, informatics, artificial intelligence, design and manufacturing journals sponsored by professional societies. Dr. Lee currently works in the School of Industrial and Systems Engineering of Georgia Institute of Technology, Atlanta, Georgia, USA as a visiting professor.

Junsu Jung got his M.S. degree at the Department of Industrial and Information Systems Engineering at Soongsil University in 2006. He has been developing a variety of software and information systems based on OOD and framework technique for more than ten years. He has interests in software development for productivity, web framework, component-based development, software architecture, etc.