

Framework and Architectural Style Metrics for Component Based Software Engineering

R. Thirumalai Selvi, Dr. N. V. Balasubramanian, George T.Manohar

Abstract— The component-based development brings many advantages such as shorter time to market and lower prices. These advantages are especially attractive for the customers, which often do not recognize the risks of lower reliability, possible problems with maintenance, etc. Many software companies are forced to use imported components in their products, but are not able to keep the development process under control. The component-based development is still a process with lot of problems, not well defined either from theoretical or practical point of view. The lack of knowledge is probably the biggest problem and the need of component-based software engineering (CBSE) is urgent. One of the important issues studied in SA is that of recurring architectural patterns and idioms – or architectural style. CBSE and SA are clearly related, and the importance of architectural issues in CBSE is now widely recognized.

However, since their motivations differ (and therefore their criteria for what should and should not be viewed as a component), the relationship between the two topics may not be straightforward. This was a motivation to a company and the software developers to a quantitative approach of calculating the framework metrics for CBSE from the case study given by the industry people.

Index Terms— ASP, COTS, Framework, MOODLE, PHP, SA

I. INTRODUCTION

There is a high trend of using components, especially COTS (commercial off the shelf) in software development. Both customers and producers share the enthusiasm in CBD approach because of the obvious advantages: The development time dramatically decreases, the usability of the products increases, the production costs usually decrease, and so on. Indeed, in many domains significant improvements in efficiency of development have been achieved. However, an inclusion of components, on which the producer does not have a complete control over, increases the risk of getting unexpected results. Even good components can corrupt a good product if they are managed in a wrong way. In some domains, such as industrial process controls, this risk is unacceptable, and additional measures are required to minimize the risks. One of the problems is a lack of established procedures and, in

R. Thirumalai Selvi is a senior lecturer of Dept. of Computer Applications, Velammal Engg. College, Chennai 600066, India. 1Ph:91-44-26591537,91-44-26591507,Fax:91-44-26590005, Email: sarasselvi@yahoo.com.

Dr. N. V. Balasubramanian is a professor of Dept. of Computer Science & Engg., RMK. Engg., College, Chennai 601206.

George T.Manohar is a professor of Dept. of Electrical, Eng, IIT Madras, Chennai 60006,India

general, a lack of knowledge of CBSE. To highlight the problems and to see which is the primary interest for industry and academia, this paper discuss about architectural metrics for CBSE .

II. COMPONENT SOFTWARE ARCHITECTURE

Architectural Styles in CBSE and software architecture (SA)[7] are separate but related topics in software engineering research and practice. CBSE focuses on the realization of systems through integration of pre-existing components. SA is concerned with the high-level organization and structure of systems in general.

A. Architectural styles in component based software engineering

One of the important issues studied in SA is recurring architectural patterns and idioms or architectural styles. CBSE and SA are clearly related, and the importance of architectural issues in CBSE is now widely recognized. However, since their motivations differ (and therefore their criteria for what should and should not be viewed as a component), the relationship between the two topics may not be straightforward. This report investigates this relationship with particular focus on architectural styles and component-based technologies such as DCOM and JavaBeans.

1) Architectural styles for Traditional Framework(ASP)

ASP stands for Active Server Pages. It is a part of Microsoft's IIS web server that enables (VBScript and/or JavaScript) scripting language embedded within HTML pages. ASP uses server-side scripting to dynamically produce web pages that are not affected by the type of browser the web site visitor is using.

With the help of ASP we can do the following things like,

- Customize a Web page to make it more useful for individual users
- The advantages of using ASP instead of CGI and Perl, are those of simplicity and speed
- Provide security since your ASP code can not be viewed from the browser
- Clever ASP programming can minimize the network traffic

When a browser requests an ASP page, the web server interprets any ASP contained within the web page before sending the HTML produced to the browser. In this way, the entire ASP is run on the web server and no ASP will ever be passed back to the web browser.

2) Architectural styles for PHP:

PHP (PHP:Hypertext Preprocessor) is a reflective programming language originally designed for producing dynamic web pages.[1] PHP is used mainly in server-side scripting, but can be used from a command line interface or in standalone graphical applications. Textual User Interfaces can also be created using ncurses.

An example of how an architectural style is implemented with components is given below.

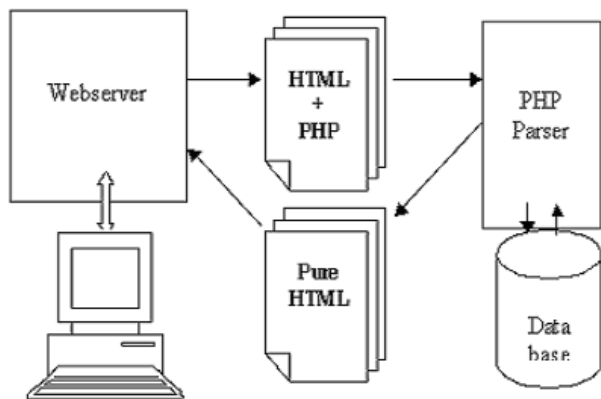


Figure 1 Interaction between Webserver and PHP Parser

PHP 5 included new features such as

- Robust support for Object-Oriented Programming
- The PHP Data Objects extension, which defines a lightweight and consistent interface for accessing databases
- Performance enhancements taking advantage of the new engine
- Better support for MySQL through a completely rewritten extension
- Embedded support for SQLite
- Integrated SOAP support
- Data iterators
- Error handling through Exceptions

3) Architectural styles in MOODLE:

Moodle (Modular Object-Oriented Dynamic Learning Environment) is an open source application built in PHP and MySQL. Moodle claims that 'external web applications can be linked in with data passed to them', which could be an opportunity to link elements of Moodle to the ELF. Moodle is a course management system (CMS) - a free, Open Source software package designed using sound pedagogical principles, to help educators create effective online learning communities.

The components of Moodle are called 'activity modules' and could be useful in informing the expansion of the ELF which is currently not so activity-based. The popularity of Moodle is partly because of the way it draws students in, and partly because it is relatively easy to install and run, not requiring any Java or .NET skills. Even though it is not a framework, it is relevant to ELF developers because its functionality has captured the imagination of many teachers and learners. Moodle is open source and therefore could be adjusted to work with web services.

The open source (OS) nature was a bonus. With the code at our disposal we could develop new functions to match our needs and the needs of other Higher Education Institutions. We also felt safe with a Java-based system as it is robust and scalable.

The below figure shows all the various components.

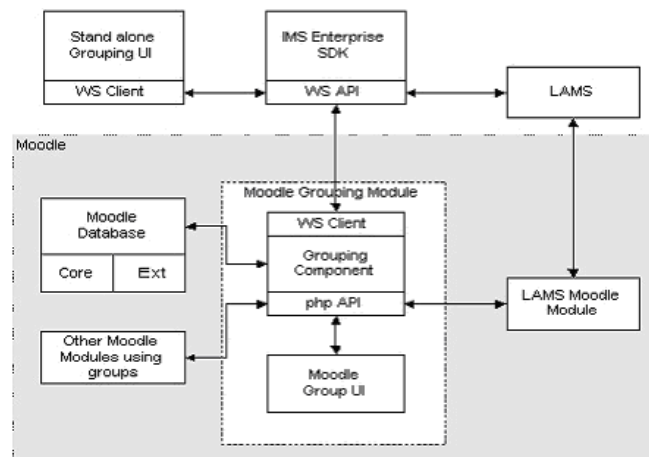


Figure 2 Architecture diagram for MOODLES

Moodle Database

This consists of the core (default) Moodle database and some extension tables. The Moodle Grouping Module will require some extension tables to be created, so that the recursive nature (ie groups can be within groups) of IMS Enterprise can be stored. The core and additional database tables are shown below:

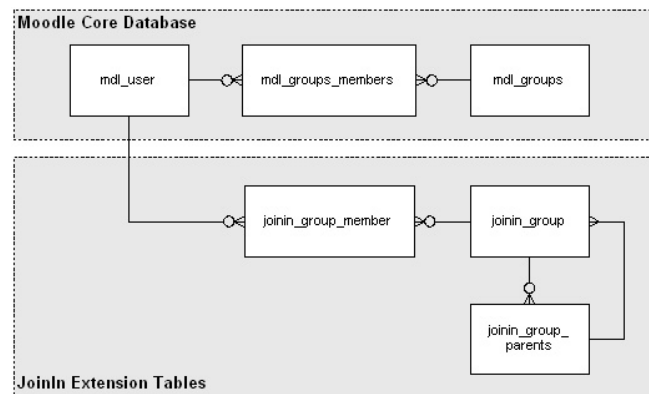


Figure 3 Diagram for MOODLES database

The web application platform that runs on most platforms is PHP combined with MySQL, and this is the environment that Moodle has been developed in (on Linux, Windows, and Mac OS X). Moodle also uses the ADODB library for database abstraction, which means Moodle can use more than ten different brands of database. Code reuse is instead achieved by libraries of clearly-named functions and consistent layout of script files.

Moodle has a number of features that are modular, including themes, activities, interface languages, database schemas and course formats. This allows anyone to add features to the main codebase or to even distribute them separately.

Moodle is not a Content Management System. If we want to build a web site, and have the ability to do easy page updates, and add-in modules for your visitors, then you should use something like one of the Nuke series (PHPNuke, PostNuke, etc), PHPWebsite, Mambo, Etomite, Joomla, Exponent, Typo3 or one of the many other portal and content systems that are now available in open source systems.

Although it is possible to create a general purpose website using Moodle, it is not optimised for this purpose, but rather for Learning and Teaching, a task with different priorities and requirements. Having said that, there is now a CMS add-on available for Moodle to allow simple web-site construction as part of a moodle site.

One thing Moodle does is keep all files for one course within a single, normal directory on the server. This would allow a system administrator to provide seamless forms of file-level access for each teacher, such as Appletalk, SMB, NFS, FTP, WebDAV and so on.

III. COMPARATIVE CASE STUDY OF VARIOUS FRAMEWORKS USING METRICS:

Metrics like PHP,ASP and MOODLES were collected from various industries like Tenth planet , Alopa Labs and also from various students. For this we have taken several static and dynamic web sites and their URL are given in the appendix.

TABLE 1 METRICS TABLE FOR ASP

Sl.No	Metrics	Weightage (max 10)
1.	Database connection, recordset, Table manipulation	8
2	Session tracking	7
3	Email object	6
4	Loops	6
5	Cookies/Hidden formFields	7
6	Cascading Style Sheet	5
7	Scripting Language Menus/buttons/events	6
8	HTML Table design	3
9	HTML Forms (get/post)	4
10	Hyper media/Hyper links	3
11	Images	3
12	Frame & Frameset	2
13	Web Contents	5

TABLE 2 METRICS TABLE FOR PHP:

Sl.No	Metrics	Meeting universe .com	Namb co.co m	Vya br.c om	T.d.A cade my
1	Platform Neutral	8	7	6	7
2	Session Tracking,	7	8	7	8
3	Creating Record Set,	8	6	5	7
4	Manipulating Tables	6	8	6	8
5	Database Connection	8	5	7	8
6	Request/Response	7	4	4	6
7	Email object	3	6	7	8
8	Hidden Form fields	5	5	4	6
9	CSS	7	6	8	8
10	Cookies	4	7	5	6
11	Contents	7	5	7	8
12	Content Management system	8	7	5	8
13	Scripting Language	7	7	8	7
14	HTML pages	5	8	7	8
15	Media files	4	3	4	8
16	Security	8	7	6	8

TABLE 3 METRICS TABLE FOR MOODLES:

Metrics	Chipkidz.com/ E-learn
Platform Neutral	8
Session Tracking	8
Creating Record Set	8
Manipulating tables	8
Database Connection	8
Request-Response	9
Email object	9
Hidden form fields	7
CSS	7
Cookies	7
Contents	8
Content Management System	10
Scripting Language	8
HTML Pages	9
Media Files	9
Security	9

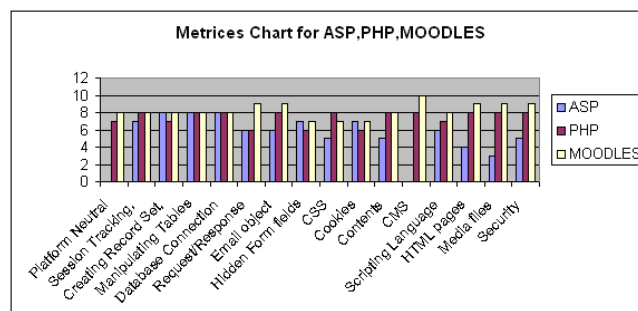


Figure 4 Metrics Chart for ASP, PHP, MOODLES

IV. CONCLUSION

This short paper reports on an investigation of the relation between software architecture and component-based software engineering, especially of the role of architectural styles in the context of systems based on standard components. Web applications using these server programs namely ASP or PHP, PHP is considered to have good solution oriented metrics than ASP language. Instead the PHP language is platform independent and user-friendly language. As it compared with MOODLES the CMS framework has more kind of metrics. The above chart shows the web application developed with the MOODLES framework that has a good optimization rather it is developed with a traditional framework.

More over it depends on the contents of the web site and by the programming. Some more static web sites would contain more metrics, if it has created with traditional framework. But by considering the dynamic web sites that should be developed CMS framework. So that it would offer more metrics and the web site is more optimized.

REFERENCES

- [1] Mary Shaw and David Garlan, Software Architecture: Perspectives on an Emerging Discipline. Prentice Hall, 1996.
- [2] Component-Based Software Engineering State of the Art, Mälardalen University, Västerås, Sweden, 2000
- [3] Christine Hofmeister, Robert Nord, and Dilip Soni, Applied Software Architecture. Addison-Wesley, 1999.
- [4] E. W. Dijkstra, "The Structure of THE-Multiprogramming System." Communications of the ACM, Vol. 11, No. 5, May 1968.
- [5] D. L. Parnas, "On the Criteria To Be Used in Decomposing Systems into Modules." Communications of the ACM, Vol. 15, No. 12, December 1972.
- [6] Dewayne E. Perry and Alexander L. Wolf, "Foundations for the Study of Software Architecture." ACM SIGSOFT Software Engineering Notes, Vol. 17, No. 4, December 1992.
- [7] D. Garlan, R. Allen, and J. Ockerbloom, "Architectural Mismatch or Why it's hard to build systems out of existing parts." Proceedings of 17th International Conference on Software Engineering, Seattle Washington, April 1995.
- [8] Metrics in Component Based Software Engineering, 2nd International Functional Sizing Summit (IFPUG), Vancouver, Canada. April 22nd - 26th, 2007
- [9] Analysing The Problem And Solution Oriented Metrics For Traditional And Component Based Frameworks For Web Applications, Developer IQ Asia's #1 Software Technology Magazine, Vol.7, No 5, May 2007, Pg no's 46-54, by Mrs.R.Thirumalai Selvi, Senior Lecturer, Velammal Engineering College, Mr. George T. Manohar, Professor, Dept. of Electrical Eng., IIT Madras, Dr.N.V.Balasubramanian, Professor, Dept. of CSE, RMK. Engg., College, Ch.
- [10] Metrics in Component Based Software Development using Frameworks and Middleware, i-manager's Journal on Software Engineering, Vol 1, No 1, No 4, April-June 2007, 12-18, by Mrs.R.Thirumalai Selvi, Senior Lecturer, Velammal Engineering College, Mr. George T. Manohar, Professor, Dept. of Electrical Eng., IIT Madras, Dr.N.V.Balasubramanian, Professor, Dept. of CSE, RMK. Engg., College, Ch.
- [11] Problem And Solution Oriented Metrics For Web based Applications, i-manager's Journal on Software Engineering, Vol 2, No 1, July-September 2007 by Mrs.R.Thirumalai Selvi, Senior Lecturer, Velammal Engineering College, Mr. George T. Manohar, Professor, Dept. of Electrical Eng., IIT Madras, Dr.N.V.Balasubramanian, Professor, Dept. of CSE, RMK. Engg., College, Ch.
- [12] Mary Shaw and Paul Clements, "A Field Guide to Boxology: Preliminary Classification of Architectural Styles for Software Systems." Proceedings of COMPSAC'97, 21st Int'l Computer

- Software and Applications Conference, Washington, D.C., August 1997.
- [13] Clemens Szyperski, Component Software: Beyond Object Oriented Programming. ACM Press Books, 1997.
- [14] Alan W. Brown and Kurt C. Wallnau, "The Current State of CBSE." IEEE Software, Vol.15, No. 5, September 1998.
- [15] Alan W. Brown and Kurt C. Wallnau, "Engineering of Component-Based Systems." Second IEEE International Conference on Engineering of Complex Computer Systems, October 1998.