Arciniegas Herrera, José Luis; Fernández de Valdenebro, María Verónica; Hormiga Juspian, María
Amparo; Tulande Arroyo, Aleyda; Collazos Ordoñez, Cesar Alberto

Architectural patterns regarding web application domain usability

Architectural patterns regarding web application domain usability

Patrones arquitectónicos sobre usabilidad en el dominio de las aplicaciones *web*

José Luis Arciniegas Herrera[1], María Verónica Fernández de Valdenebro[2], María Amparo Hormiga Juspian[3], Aleyda Tulande Arroyo[4] y Cesar Alberto Collazos Ordoñez[5]

ABSTRACT

Usability is one of the quality characteristics having the greatest impact on final user acceptation in a web application context. However, the relationship between usability and software architecture has not been completely identified and characterised. Some work has found some relationships between usability and architecture but not between architectural patterns. This paper's main contribution lies in its identification of architecture patterns and how they have influenced usability within the context of web applications. This study is useful for architects and new product designers regarding the early identification of possible problems and/or advantages (at architectural level), using some of them in a web application. Additionally, architects could identify weaknesses in initial designs thereby increasing final user satisfaction. The study could be useful when a web application is being evaluated, since detecting possible usability problems or weaknesses implies that it should be improved. Consequently, some changes would be made to the application, both in its interfaces and at architectural level. In this case, the problem can be related and the most suitable solution be suggested in terms of patterns. The impact of this work's architectural pattern has been validated on two Colombian small- and medium-sized enterprises which have exercised some of the proposed patterns and verified their influence on usability properties.

Keywords: software architecture, usability, web application, patterns, SME.

RESUMEN

En el contexto de las aplicaciones web, la usabilidad es una de las características de calidad con gran impacto en la aceptación por parte del usuario final. Sin embargo, la relación entre la usabilidad y la arquitectura del *software* no ha sido totalmente identificada y caracterizada. Algunos trabajos han encontrado algunas relaciones de la usabilidad con la arquitectura pero no con patrones arquitectónicos. La principal contribución de este artículo es la identificación de los patrones arquitectónicos y en qué medida éstos impactan en las propiedades de usabilidad en el contexto de las aplicaciones *web*. Este estudio tiene dos utilidades principales: la primera, para los arquitectos y diseñadores de nuevos productos, en la identificación temprana (a nivel de arquitectura) de posibles problemas o ventajas al usar algunos de ellos en una aplicación *web*; además, los arquitectos podrían identificar debilidades en el diseño inicial, incrementando la satisfacción del usuario final. Y la segunda, cuando una aplicación *web* está sido evaluada desde el punto de vista de la usabilidad, la detección de posibles problemas o debilidades implica que ésta deba ser mejorada, lo que conlleva a realizar cambios en la aplicación, en las interfaces y a nivel de su arquitectura, en este caso, los patrones permiten relacionar el problema y sugerir la solución más adecuada. El impacto de los patrones arquitectónicos estudiados en este trabajo han sido validados con dos pequeñas y medianas empresas (PyMes) colombianas, las cuales han ejercitado algunos de los patrones propuestos verificando su influencia en las propiedades de la usabilidad.

Palabras clave: arquitectura de *software*, usabilidad, aplicación *web*, patrones, PyMes.

Introduction

Software architecture success has a direct relationship with software quality characteristics (Bosch, 2000). Usability (Ferré *et al.*, 2001) is one of the critical software quality characteristics in web applications because it has a strong relationship with the final user. Web applications have a complex set of requirements (Valderas, 2008) and extensive use of interaction and many alternatives to present the same content to different users in different contexts

(Montero *et al.*, 2004). However, nowadays there is a gap between architectural patterns and their possible impact due to usability. An architectural pattern usually solves functional aspects; software quality characteristics have sometimes been taken into account (Arciniegas, 2006). Usability is frequently analysed without looking at its implications for architecture. Some works in the literature have thus been studied in an attempt to find such relationship. The STATUS project (Folmer, 2005) perhaps presents the major contribution as some usability properties have been found by merging usability with software architecture. STATUS usability properties

have been taken in this work, as the starting point for finding the relationship with architectural patterns.

Once the join point (patterns and usability) had been found, efforts were made to measure the potential effect when a web application used a specific pattern. This work thus focused on analysing each of the most-used patterns. Obviously, the effect of patterns could be positive or negative (i.e. the pattern could enhance web application usability properties). This work has proposed a usability scale for potential effects; the effects were theoretically proposed from information, pattern descriptions, attributes, constraints, advantages, etc. However, the usability scale has been refined through a heuristic study resulting from exercising the scale in real web application small- and medium-sized enterprise (SME) development (Batista and Figueiredo, 2000) and expert recommendations.

Usability and its properties

There are several definitions for software usability; some of the most well-known have been proposed by ISO/IEC standards 9241-11 (1998), 9126-1 (2001) and 25010 (2007). However, these definitions do not have enough scope or precision; a new definition has thus been proposed:

Usability is software's ability for a determined user to measure specific objectives with effectiveness, efficiency and satisfaction (attracttive, understood and learned) within a context of specified use.

Once usability has been defined, usability properties should be considered to achieve a connection with architectural patterns. Only the STATUS project (Folmer, 2005) has identified this relationship in the literature. Usability properties form the bridge between usability attributes and architectural patterns, in agreement with the STATUS project. Such properties have been adopted and refined in this paper and have been based on Molich and Nielsen's heuristic principles (1993). The properties considered were: feedback, error management, consistency (visual, functional and evolutionary), guidance and orientation, minimising cognitive load, explicit user control, natural mapping (predictable, user language and navigation facility), accessibility (disabilities, multichannel and internationalisation) and adaptability (user experience, customisation and rememberability).

Web applications' architectural patterns

A large set of architectural patterns are currently available; some of them are highly abstract for specific purposes (Velez and Arciniegas, 2009) and others have a fine degree of sophistication, as in that presented in Gamma et al., (1995). The work domain was a web application in this paper; Fowler et al., (2002) and Buschman et al., (1996) presented a good selection of patterns. However, not all of them have an impact on usability properties. This section describes the main web application architectural patterns having special incidence on usability properties. A classification has also been proposed for guiding architects and developers in how the patterns can be used in the web application domain. A complete description of the process has been presented by Arciniegas et al., (2009). However, taxonomy is not just for organisation for this paper because patterns have been focused on different abstraction levels or some part of a web application solution.

Classification took client/server architecture common multitier partition into account: *presentation, business and data*. Classification location has not been clear in some patterns because, sometimes, the border between the basic partition could not be

well defined. The proposed taxonomy has been based on the greatest use of patterns in real web applications and their documentation. Abstraction level plays a relevant role when taking decisions and a sub-classification has thus been proposed. Figure 1 presents the proposed classification.
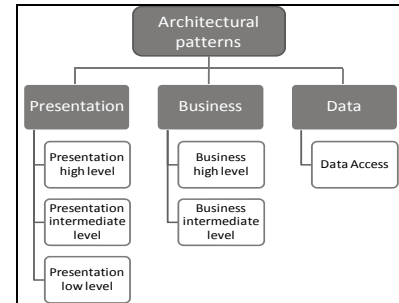

Figure 1. Web application architecture classification

Presentation

Patterns solving problems in presentation level, graphical user interface (GUI) and view controllers. GUI design is a complex task because a correct combination of elements is required both in quantity and space for intuitive information presentation:

-**High level presentation (HLP)**: Some patterns focus on user interface but have a high level of abstraction, i.e. the focus lies in trying to disengage a user interface from business and data elements. The following patterns have been located in HLP; model view controller (MVC) and presentation abstraction control (PAC) patterns;

-**Intermediate level presentation (ILP)**: Other patterns are very focused on GUI but not on a detailed level of interface design. These patterns have been grouped in this category as being: application controller, template view, transform view, two-step view, page controller and front controller. Another ILP group concerns the navigation patterns proposed by Fernandez et al., (2008); and

-**Low level presentation (LLP)**: LLP group patterns are focused on graphical details and interactions such as: information demand, information interaction decoupling, behavioural grouping, behavioural anticipation and process feed-back. Another LLP group concerns the interaction patterns proposed by Fernandez et al., (2008).

Business

A set of patterns specialising in web application rules and calculations:

-**High Level Business (HLB)**: Some patterns are focused on rules and calculations but have a high level of abstraction, i.e. focusing on business logic structure. Layer patterns have been located in BHL; and

-**Intermediate Level Business (ILB)**: Other business logic patterns are specialised depending on their solution context. This group of patterns has been grouped into: blackboard, micro-kernel, middleware and reflection.

Data

The following patterns are flow and data management intensive; however, no data structure has been considered depending on the

type of database chosen (object-orientated, relational-orientated or a mixture of both). Other patterns proposed by Fowler *et al.*, (2002) concern data domain. Some examples would be data mapping, row data entrance, table data entrance and active register. However, none of them have a special effect on usability and they have thus not been considered. Only data access patterns have been studied for this work.

-**Data Access**: These patterns focus on how to access from business logic to data store in a database. Transaction script, domain model and table module patterns have been considered.

The previous taxonomy provides a reference point for architects. Architects can use the taxonomy to identify common architectural patterns' characteristics, such as discard architectural patterns having no relationship from a usability point of view.

Web application pattern usability analysis

An impact scale table should first be defined to measure the patterns' most precise incidence on usability properties. Usability properties are a subjective aspect as the final user is usually the evaluator in the specific context of use. No impact usability scale has been documented in the literature and thus this paper defines a final user perception qualitative scale. Table 1 presents the usability impact scale, A, M and B being three levels of impact (high, medium and low, respectively). NI means that the pattern had no influence on usability property. Letters have been associated with "+" or "-" symbols, meaning positive or negative impact. "A+" thus means high positive impact (i.e. the pattern allows usability conditions), "M+" means medium positive impact (i.e. patterns may allow usability conditions), "B+" means low positive impact (i.e. the pattern could support usability conditions), "A-" means high negative impact (i.e. the pattern does not allow usability conditions), "M-" means medium negative impact (i.e. the patterns may not allow usability conditions), "B-", means low negative impact (i.e. the pattern could not support usability conditions).

Table 1. Usability impact scale

Symbol	Degree of impact
A+	High Positive
M+	Medium Positive
B+	Low Positive
NI	No Influence
B-	Low Negative
M-	Medium Negative
A-	High Negative

Each pattern being considered was analysed once the usability impact scale had been defined. In some cases some patterns had relevant usability impact and in other cases their influence was minimal; only patterns having a significant impact were thus selected.

For example, the MVC pattern had an "A+" in consistency properties because MVC clearly divided its elements into model, view and controller. Model and controller allow better functional consistency (clear location of business logic) and evolutionary consistency (easy to increase or adapt new functionalities). The view allows clear consistency in visual interfaces. Tables 2, 3 and 4 summarise the main results. An extended analysis has been presented in Fernandez *et al.*, (2008).

Validation

The usability scale has been validated in some Colombian software development (SD) SMEs. Such enterprises are especially attractive

because most Colombian SD enterprises are SMEs. However, SD SMEs have some special conditions. SMEs generally prioritise short-term functional requirements but have frequent problems regarding their software products' quality. In the worst cases, SMEs have not formalised any development process and have poor information about the role of architecture during software development. This paper also provides some basic elements from an architectural point of view which SMEs could take as reference in future developments.

IKERNELL Software Applications and INPUT Technologies Ltda (both SD SMEs) were used for validating the material. They are web application developers having some products on the market. Their products were analysed and assessed with a group of heterogeneous people (students having some knowledge about usability, developers, engineers, architects, final users etc.).

ModAvi was analysed in the case of IKERNELL Software Applications; this is an agricultural service for poultry farmers, forming part of a web application for integral agricultural business management. It is a service which should be improved in future versions. A new version was implemented following usability assessment results where presentation patterns were used to increase its usability. The improved version was assessed with the same group of people and its results were strongly enhanced.

A new requirement was analysed with INPUT Technologies Ltd to propose a new module guaranteeing usability for a web application called "Compromiso." This is a software tool based on Colombian ISO 9000 for supporting quality management. The new module manages non-conformance with entrepreneurial standards. A new application was implemented taking the presentation patterns proposed in this paper into account. Business and data patterns could be used in future versions; however, these modifications will result in profound changes in the structure of the current Compromiso application. INPUT technology developers are considering other patterns in subsequent improvement.

Conclusions and future work

Architectural patterns have a direct influence on usability properties; web application usability has been enhanced by using them. This paper has presented some architectural patterns' effect on usability properties; it could be used as a reference point for assessing web applications or taken into account when improving usability properties when deficiencies have been detected.

The results of this paper have been successfully tested in SD SMEs where some of their products have been improved from a usability point of view by using the patterns suggested here.

This paper's results should be considered as being a first approach. Other patterns could also be studied; the relationship between selected patterns could also be studied and their usability properties could be adjusted. Extra analysis should be carried out involving more enterprises putting the lessons learned here into practice.

Acknowledgements

Table 2. Relationship between presentation pattern and usability properties

Usability properties Pattern	Provide feedback	Error management	Consistency						Natural mapping			Accessibility			Adaptability		
			Visual	Funtional	Evolutionary	Guidance and orientation	Minimize cognitive load	Explicit user control	Predictable	User language	Navegation facility	Disabilities	Multichannel	Internationalization	User experience	Customization	Rememberability
MVC	M+	M+	A+	A+	A+	B+	B+	M+	M+	M+	M+	M+	M+	M+	M+	M+	M+
PAC	B-	B-	A+	A+	A+	B+	B+	B+	B+	NI	M+	B+	A+	A+	M+	A+	B+
Page controller	B+	A+	M+	M+	M+	B+	M+	M+	NI	NI	M+	B+	B+	B+	A+	A+	B+
Front controller	B+	M+	B+	B+	B-	NI	B+	M-	NI	NI	B-	B+	B+	B+	M+	M+	M+
Application controller	A+	A+	M+	M+	B+	A+	A+	A+	A+	NI	A+	M+	B+	B+	NI	NI	A+
Template view	NI	M+	NI	NI	B+	NI	NI	NI	NI	NI	NI	NI	NI	NI	B-	NI	B-
Transform view	NI	M+	NI	NI	M+	M+	B+	NI	NI	NI	B+	B+	B+	NI	B+	M+	B+
Two step view	NI	NI	A+	M+	B+	B+	NI	NI	B+	NI	B+	B-	B+	B+	NI	NI	M+
Link as a Relationship View	A+	B+	M+	A+	NI	A+	A+	M+	A+	NI	A+	M+	NI	NI	NI	NI	B-
Navigation Observer	M+	M+	NI	NI	NI	NI	A+	A+	NI	B+	A+	NI	NI	NI	B+	B+	A+
Node as a single Unit	NI	NI	M+	M+	NI	NI	A+	NI	B+	M+	A+	NI	NI	NI	M+	NI	NI
Node Creation Method and Link Creation Method	NI	NI	B-	M+	M+	NI	B-	M+	NI	NI	M+	NI	NI	NI	M+	M+	B-
Navigational Context	NI	NI	M+	M+	NI	M+	M+	A+	NI	M+	A+	NI	NI	NI	NI	NI	A+
Active Reference	M+	M+	M+	NI	NI	A+	NI	M+	NI	NI	A+	NI	NI	NI	NI	NI	A+
Information demand	A+	M+	A+	A+	M+	A+	A+	A+	A+	M+	A+	B+	B+	B+	NI	B+	B-
Information interaction decoupling	M+	NI	M+	M+	M+	NI	A+	NI	B+	M+	M+	NI	NI	B+	NI	B+	B+
Behavioral grouping	NI	NI	A+	A+	NI	A+	A+	A+	M+	B+	A+	NI	NI	NI	A+	M+	A+
Behavioral anticipation	A+	A+	M+	NI	NI	A+	A+	NI	NI	NI	NI	B+	NI	NI	NI	NI	NI
Process feed-back	A+	M+	M+	NI	NI	A+	NI	NI	B+	NI	NI	B+	NI	NI	NI	NI	NI
Interaction*																	

 * see Fernández et al, 2008

Table 3. Relationship between business pattern and usability properties

Usability properties Pattern	Provide feedback	Error management	Consistency						Natural mapping			Accessibility			Adaptability		
			Visual	Funtional	Evolutionary	Guidance and orientati	Minimize cognitive loa	Explicit user control	Predictable	User language	Navegation facility	Disabilities	Multichannel	Internationalization	User experience	Customization	Rememberability
Layers	B+	B+	M+	M+	M+	NI	NI	NI	B+	NI	NI	M+	M+	M+	NI	B+	NI
Blackboard	B-	A-	M-	A-	A-	M-	NI	NI	B-	NI	M-	M+	M+	M+	B+	B+	B-
Micro-kernel	B+	B+	B+	A+	B+	NI	NI	NI	NI	NI	NI	M+	M+	M+	M+	B+	M+
Middleware	M+	B+	NI	B+	B+	NI	NI	NI	NI	NI	NI	B+	B+	B+	NI	M+	NI
Reflection	B+	M-	NI	M+	M+	B+	NI	B-	M+	M+	M+	A+	A+	A+	A+	A+	A+

Table 4. Relationship between data pattern and usability properties

Usability properties Pattern	Provide feedback	Error management	Consistency						Natural mapping			Accessibility			Adaptability		
			Visual	Funtional	Evolutionary	Guidance and orientati	Minimize cognitive loa	Explicit user control	Predictable	User language	Navegation facility	Disabilities	Multichannel	Internationalization	User experience	Customization	Rememberability
Domain model	M+	M+	B+	M+	M+	B+	B+	NI	NI	NI	M+	B+	B+	B+	M+	M+	B+
Table module	B+	B+	B+	M+	M+	B+	B+	NI	NI	NI	M+	NI	NI	NI	B+	B+	NI
Transacción Script	M+	B-	NI	NI	NI	NI	NI	NI	NI	NI	NI	NI	NI	NI	B-	B-	NI

References

Arciniegas, J., Contribution to quality-driven evolutionary software development for service oriented architecture., Universidad Politécnica, Madrid, Ph.D. degree thesis, 2006.

Arciniegas, J., Fernández, V., Tulande, A., Hormiga, A. Urbano, F., Collazos, C., Proceso de requerimiento y análisis para la definición de la arquitectura desde la perspectiva de usabilidad para el desarrollo de una aplicación en la Web., Revista Avances en Sistemas e Informática, Vol 6, No. 2, Septiembre, 2009, pp. 205 - 210.

Batista, J., Figueiredo, A., SPI in very small team: a case with CMM., Software process improvement and practice, DOCIS Documents in computing and information science, 5 (4), 2000, pp. 243-250.

Bosch, J., Design and use of software architectures: adopting and evolving a product line approach., Pearson Education, Addison-Wesley, 2000.

ISO 25010., International Standard. Software product Quality Requirements and Evaluation (SQuaRE): Quality model., 2007.

ISO 9241-11., International Standard. Ergonomic requirements for office work with visual display terminals (VDTs)-Part 11: Guidance on usability., 1998.

ISO/IEC 9126-1., International standard, software engineering-product quality-Part 1: Quality model., 2001.

Fernández, V., Hormiga, M., Tulande, A., Marco de referencia centrado en la arquitectura para la mejora de características de usabilidad en el desarrollo de aplicaciones web construidas por MIPYMES., Undergraduate thesis, Universidad del Cauca, 2008.

Ferré, X., Juristo, N., Windl, H., Constantine, L., Usability basics for software developers., IEEE Software, Vol. 18, No. 11, 2001, pp. 22-30.

Folmer, E., Software architecture that supports usability., Ph.D. Memoria de la tesis doctoral, Para grado de Doctor en Mate-máticas y ciencias naturales en la Universidad pública Groningen, 2005.

Fowler, M., Rice, D., Foemmel, E., Hieatt, E., Mee, R., Stafford, R., Patterns of Enterprise Applica-tion Architecture., Addison Wesley, 2002.

Gamma, E., Helm, R., Johnson, R., Vlissides, J., Design patterns elements of reusable object-orientated software., Addison-Wesley, 1995.

Montero, S., Díaz, P., Dodero, J., Aedo, I., AriadneTool: A design tool-kit for hypermedia applications., Journal of Digital Infor-mation, 2004.

Nielsen, J., Usability Engineering., Academic Press Professional, Boston, MA., 1993.

Nielsen, J., Heuristic Evaluation., in: J. Nielsen and Mack, R. L. (eds.), Usability Inspection Methods, John Wiley and Sons, New York, N.Y., 1994.

Valderas, P., A requirements engineering approach for the development of web applications., Universidad Politécnica de Valencia, Ph.D. degree thesis, 2008.

Vélez, F., Arciniegas, J., Arquitecturas de gestión de redes y servicios de telecomunicaciones., Editorial de la Universidad Libre (Cali, Colombia), 2009.