2012 Eighth International Conference on the Quality of Information and Communications Technology

Evaluating the Usability of Mashups Applications

Emilio Insfran, Priscila Cedillo, Adrián Fernández, Silvia Abrahão
Universitat Politècnica de València
Valencia, Spain
{einsfran, icedillo, afernandez, sabrahao}@dsic.upv.es

Maristella Matera
Politécnico di Milano
Milan, Italy
matera@elet.polimi.it

Abstract—Mashups are a new-generation of Web applications whose objective is to reuse the contents and services provided by third-party components. In Web applications, usability is considered to be one of the most important quality factors. However, a usability evaluation of mashups usually relies upon their individual components. There is therefore a need for specific approaches with which to evaluate the usability of mashups by considering their specific characteristics as composite artifacts. This paper proposes a Usability Model for Mashups that can be used to evaluate the usability of mashup applications. This model breaks down the usability sub-characteristic from the ISO/IEC 25010 standard into other sub-characteristics and attributes. Metrics are eventually associated with these attributes in order to quantify and detect usability problems. The feasibility of the approach is illustrated through the evaluation of a mashup application using the proposed usability model.

Keywords; Mashups; Usability Evaluation; Usability Model; Metrics; SQuaRE

I. INTRODUCTION

Web applications have recently become the backbone of business and information exchange, and the ease or difficulty that users experience when using these systems will determine their success or failure[9]. Mashups are a new class of Web applications that combine various Web services, RSS, ATOM feeds and other data sources to produce new applications. Emerging technologies such as Web services, User Interface (UI) widget libraries, and tool-specific mashups have significantly simplified the access and reuse of such building blocks, leading to a component-oriented paradigm which is shared by several current mashup platforms [8].

Mashups have gained popularity as a result of the balance between the development cost, ease of integration of third-party components and quality of the solutions that they provide. There are some specific characteristics for mashups that make them different from traditional Web applications, such as their compositional essence. Mashups rarely have the concept of navigation since they are usually presented on an individual Web page. What is more, they are created for specific situations and are often employed for short periods of time. These specific characteristics have led to the need for a usability model that is specifically tailored to deal with these characteristics.

In this paper, we propose a Mashup Usability Model that is aligned with the ISO/IEC 25010 standard (SQuaRE)[3] and extends the Web Usability Model for traditional Web applications presented in [9] to mashups. This Mashup Usability Model is used to evaluate and improve the usability of mashup applications by considering the specific characteristics of mashups (e.g., composition issues).

This paper is organized as follows. Section 2 discusses related work, and in particular, previous quality and usability models for Web and mashup applications. Section 3 presents our Mashup Usability Model. Section 4 provides an example with the aim of showing how our Mashup Usability Model is applied to evaluate a real mashup application. Finally, Section 5 presents our conclusions and further work.

II. RELATED WORK

Several quality models for Web usability evaluation have been proposed in the last few years. These models have been defined from scratch or defined based on existing standards such as ISO/IEC 9241-11and ISO/IEC 9126-1.

Quality models for the Web that are derived from existing standards include the proposals of Olsina and Rossi [4], Calero *et al.*[6], Seffah*et al.*[5], and Moraga *et al.*[7]. Olsina and Rossi [4] proposed the Web Quality Evaluation Method (WebQEM) with which to define quality characteristics and attributes based on the ISO/IEC 9126, such as *usability*, *functionality*, *reliability*, and *effectiveness*; the Web audience's needs are also incorporated. Calero *et al.*[6] present the Web Quality Model (WQM), in which three dimensions are distinguished: *Web features* (content, presentation, and navigation); *quality characteristics* based on the ISO/IEC 9126 (functionality, reliability, usability, efficiency, portability, and maintainability); and *lifecycle processes* from the ISO/IEC 12207(development, operation and maintenance). *Organizational processes*, such as project management and reuse program are also included. With regard to new generation Web applications, Olsina *et al.*[10] updated the first version of the 2Q2U (Quality, Quality in use, actual Usability and User experience) modeling framework, intended for traditional Web applications, by considering specific features for modern Web applications (Web 2.0), e.g., *integratedness*, *beneficialness*, *communicability*, *sense of community*.

Cappiello *et al.*[8] present an approach that addresses information quality in mashups. They state that *mashup users* are not interested in how the mashup was built, but simply want the mashup application to perform as expected, without missing or badly aligned data or similar information quality problems. In other words, the user is interested in the perceived external quality. These authors define mashup quality in terms of *Component Quality*, *Composition Quality* and *Information Quality*. The authors point out that

978-0-7695-4777-0/12 $26.00 © 2012 IEEE
DOI 10.1109/QUATIC.2012.28

the composition quality category "is very peculiar for mashups since it focuses on the way components interact among them and measures the utility of the composition".

In [1] Cappiello *et al.* proposed a quality model that emphasizes the component-based nature of mashups and particularly focuses on composition quality. However, to the best of our knowledge, there are no guidelines on how to evaluate the usability of mashups by considering their intrinsic characteristics. Moreover, there are no standard-based usability models for mashups that facilitate quality evaluations of this kind of Web applications.

III. MASHUP USABILITY MODEL

In this section, we provide a brief description of the main sub-characteristics, attributes, and certain metrics of our Mashup Usability Model. This model is an adaptation and extension of the Web Usability Model presented in Fernandez *et al.*[9], which breaks down the usability concept for Web applications into the six sub-characteristics proposed in the ISO/IEC 25000 SQuaRE standard, which are in turn broken down into other sub-characteristics and attributes by considering the ergonomic criteria proposed in [2] and usability guidelines for Web development [12].We have adapted the definition of these sub-characteristics and attributes, and we have also added new ones, in order for them to fit with the mashups' nature. This task was supported by first conducting usability evaluations of samples of mashups from the Programmable Web repository with the aim of identifying which usability problems were related to compositionality issues. Our entire model, including all the sub-characteristics, attributes and their associated metrics, is available at http://www.dsic.upv.es/~einsfran/MUM.htm.

In the following, owing to space constraints, we *only* show the three sub-characteristics that were modified in our original model for the evaluation of mashups, together with the new added usability attributes that are *specific* to mashup applications. The attributes in italics are explained in greater detail.

The first sub-characteristic of the Mashup Usability Model is **Appropriateness recognizability,** which is broken down into other sub-characteristics and attributes. Table I shows an excerpt of the sub-characteristics and their specific attributes.

The *Components' disposition* attribute (1.1.4) is related to the way in which components are organized on the Web page for their visual identification. It is important to maintain a balance in the distribution of components on the Web page in order to avoid an overload of information in regions or areas, and to avoid the overlapping of components. This attribute can be measured by considering the *ratio of visible components* metric (number of fully visible components per total number of components), which refers to the components that are fully visible (with no overlapping), and the *ratio of useful regions* metric of the mashup (number of regions used on the screen per total number of regions of the screen).

TABLE I. APPROPRIATENESS RECOGNISABILITY SUB-CHARACTERISTIC

Sub-characteristic	Attribute	Meaning
1.1. Visual legibility	1.1.4 Components' disposition	Are components easy to find and recognize on the Web page?
	1.1.5 Component size	Are the sizes of the components appropriate to see and use comfortably?
1.2. Readability	1.2.1. Components' grouping coherence	Are components grouped coherently according to their purpose?
1.3. Familiarity	1.3.3. Popularity of components	Are components known, accepted and commonly used by the community?
	1.3.4. Presentation of controls in components	Are the controls in the components used as usual?
1.4. Workload reduction	1.4.4. Updating of information among components	Are the interconnected components updated automatically?

The *Component size* attribute (1.1.5) is related to the size of the components for their visualization and use. Components will be useful and *usable* to the users only if they can be easily managed and if they can display all the necessary information without any effort being made by the user. The way in which the metrics for this attribute are defined and calculated will depend on the type of component and the type of information that the components manage (e.g., a map or messages). A metric for this attribute is the *ratio of components with appropriate size* (number of components with appropriate size to visualize relevant information and controls per total number of components).

The *Popularity of the components* attribute (1.3.3) is related to the familiarity with and common acceptance of third-party components by users. For example, using the Google Maps API or another similar component from a lesser known company, usually makes the component more or less recognizable to the user. A metric for this attribute is the *ratio of well-known components* (number of well-known components per total number of components).

The *Update of information among components* attribute (1.4.4.) is related to the way in which the components synchronize their content (refresh) among themselves without the user's intervention. A metric for this attribute is the *ratio of auto-synchronized components* (number of components that automatically refresh their data per total number of components that share some data with other components).

The second sub-characteristic of the Mashup Usability Model is **Learnability.** Table II shows an excerpt of the sub-characteristics and their specific attributes for mashups.

The *Distinguishable components purpose* attribute (2.1.4) is related to the contribution of the components to the general purpose of the mashup. A metric for this attribute is the *ratio of distinguishable components* (number

of components that contribute to the purpose of the mashup and the total number of components).

The *Realization of possible component actions* attribute (2.2.3) is related to the ease or difficulty that the user experiences when carrying out the possible actions that can be realized with a component. A metric for this attribute is the *ratio of component actions easily identified* (number of components with all the possible actions easily identified per total number of components).

TABLE II. LEARNABILITY SUB-CHARACTERISTIC

Sub-characteristic	Attribute	Meaning
2.1. Predictability	2.1.4. Distinguishable purpose of components	Do components contribute to the mashup purpose?
2.2. Affordance	2.2.3. Realization of possible component actions	Do components show what they do clearly?

The third sub-characteristic of the Mashup Usability Model is **Operability**. Table III shows an excerpt of the sub-characteristics and their specific attributes for mashups.

TABLE III. OPERABILITY SUB-CHARACTERISTIC

Sub-characteristic	Attribute	Meaning
3.2. Controllability	3.2.8. Reachability of components	Can the user reach all the components in the mashup?
	3.2.9. Load of components	Are components difficult to load because of their size? Is there any restriction that makes the loading of components difficult?
3.3. Consistency	3.3.9. Correct behavior of components	Do components behave correctly according to their input data/event?
3.4. Portable Executions	3.4.1. Device independence	Is the operability of the mashup similar across platforms?
	3.4.2. Auto-adjustment of components to devices	Are components are resized according to different platforms, devices or screen sizes?
	3.4.3. Suitability for additional software required	When components need plugins, how difficult is to install them?

The *Reachability of components* attribute (3.2.8) is related to the possibility of reaching all the components in the mashup. A large number of components or inappropriate sizes may cause the components to be hidden from the user. A metric for this attribute is the *ratio of reachable components* (number of reachable components per total number of components).

The *Correct behavior of components* attribute (3.3.9) is related to the component's behavior as regards what it is expected to do as a result of its input data or event. A metric for this attribute is the *ratio of correct behavior of components* (number the components with appropriate behavior per total number of components).

In addition to the sub-characteristics defined in the Web Usability Model presented in [9], we include a new sub-characteristic specific for mashups to evaluate the components as regards their ability to be executed across platforms or devices. This sub-characteristic, *Portable Executions (3.4)*, is also shown in Table III.

The *Device independence* attribute (3.4.1) is related to the way in which the operability of the mashup is affected when used across platforms. A metric for this attribute is the *ratio of platforms with the same operability* (number of platforms with the same operability per number of platforms).

The *Auto-adjustment of components to devices* attribute is related to the way in which the mashup is presented in different screens and devices. Finally, the *Suitability for additional software required* (3.4.3) is related to the additional software that it is necessary to install in order to execute the mashup. It is sometimes necessary to install plugins that are not available for all the devices or platforms (e.g., Flash for iPhone/iPad).

IV. APPLYING THE MASHUP USABILITY MODEL

In order to illustrate the use of the Mashup Usability Model for usability evaluation, we use the Liveplasma mashup (http://www.liveplasma.com/), which has been selected as one of the most popular mashups published in the *programmableweb.com* repository. This mashup searches for music, movies and books, and the results are presented in a graph model showing the information related to a search criteria (see Fig. 1). This mashup combines the Amazon Services with YouTube as third-party components.

In order to perform the usability evaluation of a mashup application, it is first necessary to *identify and select a set of sub-characteristics, attributes and metrics* of interest from the Mashup Usability Model. Then, we *apply the metrics* to the mashup in order to obtain the values which quantify the corresponding attributes and sub-characteristics. This information enables us to *obtain a usability report* which highlights the usability problems detected and the corresponding suggestions to fix them.

Owing to space constraints, we illustrate the usability evaluation using three attributes with their metrics and rating levels, thus producing a brief usability report. The attributes selected for the evaluation of the mashup shown in Fig. 1 are: Components disposition (1.1.4), Realization of possible component actions (2.2.3), and Update of information among components attribute (1.4.4).

With regard to the *Components disposition* attribute, we apply the *ratio of visible components* metric. We have 4 components (Amazon, YouTube, graph and searcher) and we can see that the *graph* component is overlapped by the others. Upon performing the calculation of the metric, we obtain a value of 0.75 (3 components fully visible of a total of 4 components).

With regard to the *Realization of possible component actions* attribute, we apply the *ratio of component actions easily identified* metric. In this case, all the components clearly show the possible actions when moving the mouse on top of them. Upon performing the calculation of the metric, we obtain a value of 1 (4 components with all their

actions clearly identified by a total of 4 components). With regard to the *Update of information among components attribute*, we apply the *ratio of auto-synchronized components* metric. In this case, when we select a related node from the graph component or introduce a new name in the searcher component, all the information in the graph and in the Amazon titles (top left area of the mashup in Fig. 1) is auto-updated. However, the YouTube component continues to play the video related to the previous element, unless the user presses the "Solo" button (a control in the graph node), which synchronizes the YouTube video with the main content of the mashup. Upon performing the calculation of the metric, we obtain a value of 0.75 (3 components which are always auto-synchronized of a total of 4 components).

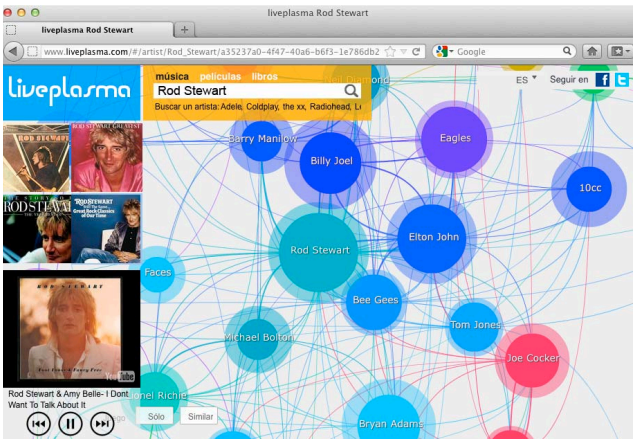

Figure 1. Liveplasma mashup

Finally, Table IV shows a brief summary of the usability report containing the information concerning the attributes and metrics evaluated, and the results. The interpretation of all the aforementioned metrics is: $0 \leq$ value ≤ 1, in which the closer the value is to 1 the better. For the purpose of this example, the following rating levels were established by a mashup designer: [$0 \leq$ value < 0.4]: major usability problem; [$0.4 \leq$ value < 0.7]: medium usability problem; [$0.7 \leq$ value < 1]: minor usability problem; and [1]: no usability problem.

TABLE IV. USABILITY REPORT SUMMARY

Attribute	Metric	Value
1.1.4.Component disposition	*Ratio of visible components* (number of fully visible components per total number of components)	3/4 = 0.75 (minor)
2.2.3 Realization of possible component actions	*Ratio of component actions easily identified* (number of components with all the possible actions easily identified per total number of components)	4/4 = 1 (no)
1.4.4Updating of information in components attribute	*Ratio of auto-synchronized components* (number of components that automatically refresh their data per total number of components that share some data with other components).	3/4 = 0.75 (minor)

V. CONCLUSIONS AND FURTHER WORK

In this paper, we have presented a Mashup Usability Model that can be used to evaluate the usability of mashup applications and as a guide to avoid usability problems throughout the mashup development process. This Usability Model breaks down the usability sub-characteristic of the ISO/IEC 25010 standard into other sub-characteristics and attributes. Metrics are eventually associated with each one of these attributes in order to quantify them and detect usability problems.

Nevertheless, we are aware that not all the usability problems can be detected based on the inspection of the mashup, since it is not possible to predict the user's behavior and preferences. However, studies such as that of Hwang and Salvendy[11] claim that usability inspections through the application of well-known usability principles to software artifacts, would be capable of discovering around 80% of usability problems.

As a further work, we plan to: define a usability evaluation process that will include detailed guidelines on how evaluators could apply our Mashup Usability Model; explore proper thresholds and aggregation mechanisms for values obtained by individual metrics; and perform analyses of the impact to discover how the attributes affect (negatively or positively) other attributes in the Mashup Usability Model.

ACKNOWLEDGEMENT

This research work is funded by the MULTIPLE project (TIN2009-13838), the Senescyt program (scholarships 2011), the FPU program (AP2007-03731), and the ERC grant SeCo.

REFERENCES

[1] C. Cappiello, F. Daniel, A. Koschmider, M. Matera, M. Picozzi: A Quality Model for Mashups In: LNCS 6757, ICWE2011, S. Auer, O. Díaz, G. Papadopoulos (Eds.), Paphos, Cyprus, pp. 137-151, 2011.

[2] J.M. Bastien, D.L, Scapin: Ergonomic Criteria for the Evaluation of Human-Computer Interfaces. TR.156. INRIA, France, (1993).

[3] ISO/IEC: ISO/IEC 25010 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models, (2011)

[4] L.Olsina, G. Rossi: Measuring Web Application Quality with Web QEM. IEEE Multimedia, Vol. 9 no. 4 (2002) 20-29

[5] A. Seffah,M. Donyaee, R.B. Kline, H.K. Padda: Usability Measurement and Metrics: A Consolidated Model. Software Quality Journal 14(2) (2006) 159-178

[6] C. Calero, J. Ruiz, M. Piattini: Classifying Web Metrics Using the Web Quality Model. Emerald Group Publishing Limited. Vol. 29, nº 3 (2005) 227-248

[7] M.A. Moraga, C. Calero, M. Piattini, O. Diaz: Improving a Portlet Usability Model. Software Quality Ctrol, Vol. 15 nº 2 (2007) 155-177

[8] C. Cappiello, F. Daniel, M. Matera, C. Pautasso, Information Quality in Mashups.IEEE Internet Computing, 14(4): 14-22 (2010).

[9] A. Fernandez, E. Insfran, S. Abrahão: Integrating a Usability Model into Model-Driven Web Development Processes. Web Information Systems Engineering (WISE), Poznan, Poland, (2009).

[10] L. Olsina, P. Lew, A. Dieser, B. Rivera: Updating Quality Models for Evaluating New Generation Web Applications. Journal of Web Engineering (to appear)

[11] W. Hwang, G. Salvendy: Number of people required for usability evaluation: the 10±2 rule. In Communications of the ACM 53(5), 130-133 (2010).

[12] Leavit, M.; Shneiderman, B.: Research-Based Web Design & Usability Guidelines. U.S. Government Printing Office, (2006).