International Journal of Advanced Computer Technology (IJACT)

ISSN:2319-7900

TOWARD USABILITY EVALUATION CRITERIA FOR WEB MASHUP MAKERS FOR END- USERS

Wael Al Sarraj, Faculty of Information Technology, Islamic University of Gaza, Palestine.

Abstract

Currently, more than two billions people access the web for various purposes. The majority are people without programming or modeling backgrounds. Part of these people also likes to create their own Web applications to meet their daily needs. Mashup Makers are tools to create such end-user's Web applications. As such, Mashup Makers could become the dominant environment for end-user development of Web applications. Several Mashup creation tools, so-called Mashup makers, exist and many of them are advertised as easy to use tools for end-users. These tools target Web users without programming background and promise that creating a Web Mashup is just a matter of a few mouse clicks. However, no profound usability evaluations have been performed to justify those claims. In this paper, we report on an initial usability study on Web Mashup makers for end-users. For this study, we have selected a representative set of Web Mashup makers for end-users, and defined a number of evaluation criteria based on the cognitive dimensions of notations framework. The results of this study are presented in this paper. We also reflect on the methodology used.

Categories and Subject Descriptors

H.1.2 [User/Machine Systems]: Human factors; D.1.7 [Visual Programming

General Terms

Experimentation, Human Factors

Keywords

Web Mashups, Usability, End-user

1. Introduction

Mashup originally referred to the practice in pop music [19] (notably hip-hop) of producing a new song by mixing two or more existing pieces. In computer technology, a Mashup is a Web application that integrates data from more than one source. A well-known example is the use of cartographic data from Google Maps to add location information to some costumer's data, thereby creating a new service that was not originally provided by either source. The most common way to develop a Mashup interactive Web application is by accessing content for Mashups via a public interface or API. This allows developers to directly feed data from databases and other sources and enable developers to build rich content applications that make information more useful to users.

How mash-up works? "The common scenario"

As shown in Figure 1 [24], the user requests to combine available data from two or more sources (from two or more API content providers). That data is made available by relevant Web protocols such as REST [20], RSS [20] and Web services [20]. The data is scraped from the output of these APIs, and then the scraped data is going to pass to the Mashup site where the logic resides, it could be server-side (dynamic content aggregation) and/or client-side scripting or both of them. The application then is rendered graphically and transfered to the client's Web browser where user interaction takes place, that mashing is usually done by a client side Web language (e.g., JavaScript [20], Ajax [20]).

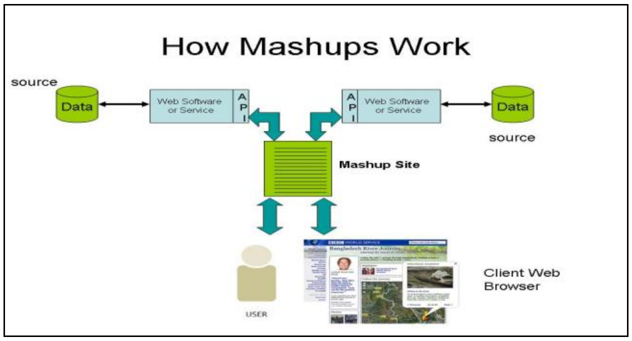


Figure 1. How Mash-ups work [24]

Several Mashup creation tools, so-called Mashup makers, exist and many of them are advertised as easy to use tools for end-users. Our research objective is to investigate the usability of Mashup makers for end-users, i.e. users without programming background. We believe that this research is very important from two aspects. The first aspect is to check if those Mashup makers indeed fulfill their promises and meet the needs of end-users. The second aspect concerns usability necessities for Mashup makers for end-users. Our investigations and studies may reveal a considerable amount of hindrances and difficulties that end-users are facing when using Mashup makers. One of the results of our usability study could be a set of minimal usability necessities for those kinds of tools. It is also our aim to develop a set of usability criteria that can be used to measure usability of existing and future Mashup makers.

The goal of this paper is to report on a first study done in the context of our usability investigation. The results if this study will be a key factor in our next research on usability of Mashup tools. Usability is an essential factor affecting the quality of web applications development environments [26]. Our ultimate goal is to perform a large usability experiment to evaluate the usability of Mashup makers. For this we need a good set of evaluation criteria for usability of Mashup makers. The study described in this paper has been set up in order to obtain the necessarily information to come to such a set of evaluation criteria.

The paper is organized as follows. In section 2, we present the results of a review performed on Web Mashups, their classification, tools for creating Mashup and their users. In this section, we also review related work. In section 3, we explain our methodology used for the usability evaluation study that we performed. In section 4, we present the results obtained from the usability evaluation study, and also discuss the results and the limitations of the findings. Section 5 concludes our paper and outlines further work.

2. REVIEW

There are two different ways to classify Mashups, regarding to their technological architecture (client-side and server-side Mashups) and to their usage (consumer, data, business, mapping, video & photo, search & shopping and news).

2.1 Server-side and Client-side Mashups

The two primary Mashup styles are server-side Mashups and client-side Mashups. As you might expect, server-side Mashups integrate services and content on the server. The server acts as a proxy between a Web application on the client, typically a browser, and the other Web site that takes part in the Mashup. In a server-side Mashup, all the requests from the client go to the server. So, in a server-side Mashup, all the work is done on the server. As an example, figure 2 shows how Java server side Mashups work [14].

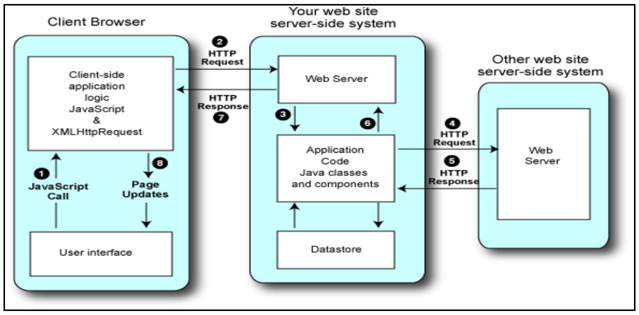


Figure 2. How server side Mashups work [14]

Client-side Mashups integrate services and content on the client. They mash up directly with the other Web site's data or functionality. For example, in a client-side Mashup, the client directly requests the other Web site. Figure 3 gives an overview of how a Java client side Mashups work [14].

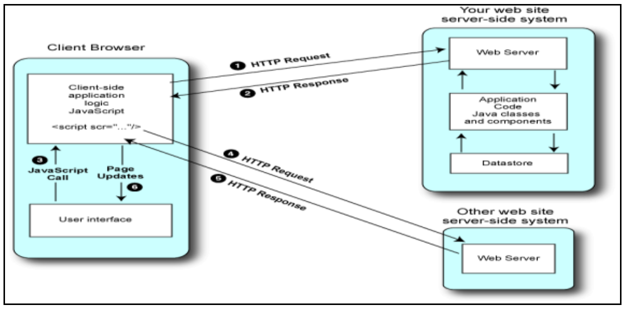


Figure 3. How client side Mashups work [14]

2.2 Classification of Mashups

Wong J. and Hong J. have developed Marmite [21], a Mashup development tool. They also published a paper on Mashups patterns [22]. Basing on a survey findings, they have classify patterns in Mashups into categories (see table 1.)

Table 1. Patterns in Mashups [22]

Category	Description
Search	Is the Mashup a search interface?
Visualization	Does the Mashup add visualization to the data?
Real-time	Is the purpose of the Mashup to allow the user to monitor or observe the original Website as a real-time data set?
Widget	Is the Mashup actually a widget for some platform like Apple’s Dashboard, the iPhone, or Google Homepage?
Personalized	Does the Mashup make use of the user’s personal information from the original Website or enable the construction of a personalized data set from the original Website?
Folksonomy	Does the Mashup make use of a tagging system or adds tagging to the original data set?
In-situ use	Is the Mashup simply a tailoring of an original Website optimized to a specific situation use?

Mashups can also be divided into many categories depending on their usage, such as: mapping, video, photo, search, shopping and news. According to programmableWeb.com [16] which is the most well-known Web site dealing with APIs and Mashup tracking on the Web, at the time of writing this paper the top Mashup tags on the Internet are ‘mapping’, ‘video’ and ‘social’ Mashup tags.

2.3 General purposes Mashups Makers

Our classification of Mashups combines both the technology and usage flavors [19]. We classify the Mashups creation tools into two main types: general-purpose Mashups makers and specific-purpose Mashups makers. Our classification is going with the end-user perspective. In general, a end-user is seeking for a general-purpose Mashup maker for quickly creating small Web applications for multiple purposes and with ease of use.

In the next sub-sections we present some of the main tools for developing general-purpose Web Mashups. All these tools have one thing in common: they make it easier to do things that, until now, only programmers could do. Each of them has quite a different approach, which makes it a very diverse set of tools.

2.3.1 Yahoo pipes

Yahoo! Pipes [23] is a visual drag and drop Mashup creation tool for fetching and merging data from different sources. It does not require knowledge of programming languages, but does require good understanding of data formats. The composition tool runs in the browser and is based on standard Web technologies. The Mashup creation area is visually divided into 3 panes: on the left contains a library that lists all functional modules that can be pulled onto the canvas; at the bottom there is a debugger area that allows checking intermediate outputs; the canvas itself is the third pane where modules are linked with the connectors or “pipes” which define the data flow (see Figure 4). Many different things are possible in Pipes. One can combine many feeds into one, then sort, filter and translate it; geo code favorite feeds and browse the items on an interactive map; or create power widgets/badges and place them on personal Web site. Pipes support a variety of output formats such as RSS, JSON [20], KML [20] and some others.

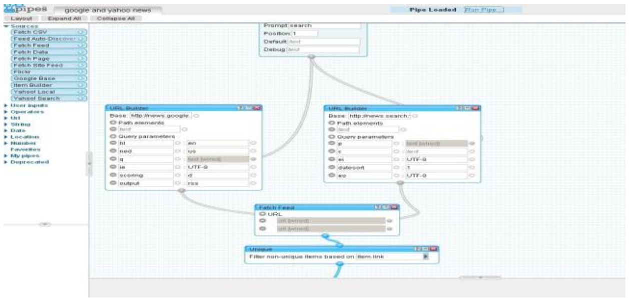


Figure 4. Yahoo pipes data Mashup tool

2.3.1 Microsoft Popfly

Microsoft® Popfly™ [12] is a Web site and tool to help people creating and sharing Web sites, Mashups, and other kinds of experiences. It has two parts: the social network, which is called "Popfly Space" and the online tool for creating different kinds of experiences, which is called "Popfly Creator." We are interested in the latter part of the service. Figure 5 shows a sample Popfly Mashup creation mode.

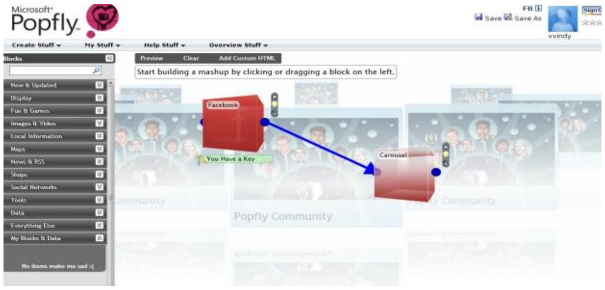


Figure 5. Popfly in the Mashup creation mode

2.3.3 Google Mashup tools

Google [6] decided on a different approach. Instead of launching an all-encompassing Mashup making application, they have offered a multitude of tools - some, like Google Mashup Maker, aimed at developers, while others, like MyMaps, can be used by anyone. Their impressive collection of APIs can be found easily by searching the Google engine, and while they don’t give many options of Mashup-creation to the layman, their openness has resulted in a huge number of great Mashups based on their services.

Google Mashup Editor is “simple” if one has developer kills and is familiar with technologies like XML tags, JavaScript, CSS [20], and HTML. A Mashup application is described in a form of high level XML [20] based language that is interpreted by the Google Mashup engine. Currently that service is in beta phase and is accessible by an invited set of people only. Google Mashup Editor [6] documentation allows classifying it as software developer tools and not an end user Mashup environment. Google Mashup editor will be excluded from our selection of Mashup makers when we compare different Mashup environment because it does not target end-users.

2.3.4 Marmite

The Marmite [21] idea is that Mashup structure comprises of sources, processors and sinks. Sources enable adding data into Marmite by querying databases, extracting information from Web pages, and so on. Processors allow to modify, combine or delete existing rows/columns (geocoding, filtering). Sinks then redirect the flow of data out of Marmite (showing data on a map, saving it to a file/Web page). Since Marmite runs as a part of the Web browser it provides easy and visual way to specify part of any Web page as a

ISSN:2319-7900

"source". It is a research project meaning that functionality and implementation quality is low compared to other tools. Marmite is implemented as a plug-in for the FireFox browser. It implements the data flow model, but the papers mention about plans to support a spreadsheet model or mixed spreadsheet and dataflow model (Figure 6).

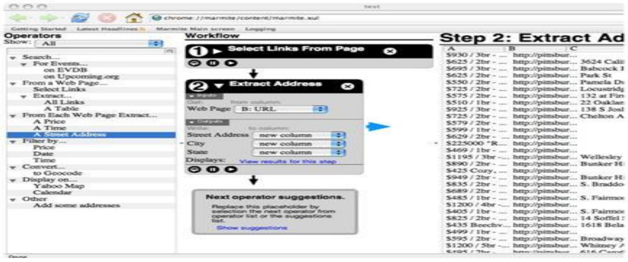


Figure 6. Marmite Mashup tool

2.3.5 Intel® Mash Maker

Intel® MashMaker [5] is an extension to an existing Web browser that allows you to easily augment the page that you are currently browsing with information from other Websites. As you browse the Web, the Mash Maker toolbar suggests Mashups that it can apply to the current page in order to make it more useful for you. For example: plot all items on a map, or display the leg room for all flights. Intel® Mash Maker learns from the wisdom of the community. Any user can teach Mashmaker new Mashups, using a simple copy and paste interface, and once one user has taught Mash Maker a Mashup, this Mashup will be automatically suggested to other users [5]. Intel® Mash Maker also relies on the community to teach it about the structure and semantics of Web pages, using a built in structure editor. There is no dedicated page on the Web where to have to go and construct the Mashup application. User has to install the toolbar in the browser and start browsing the Web. That plug-in supports multiple modes that give opportunity either to use existing Mashups or to define page structure if needed by turning your browser in a DOM explorer tool by opening new panes along with the main page (see Figure 8).

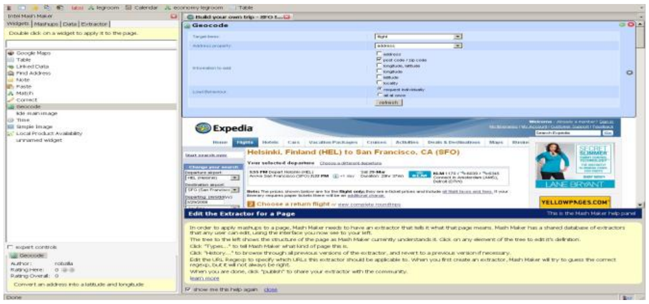

Geocode
Expedia
Helsinki, Finland (HEL) to San Francisco, CA (SFO)
Edit the Extractor for a Page

Figure 7. Mash maker is integrated in the browser and can open extra panes around the main page that used visits

The programming paradigm could be called "annotate and mix while browsing". The service is in beta and not available to a wide public at the moment of writing.

2.3.6 Dapper

Dapper [4] stands for data mapper. The main purpose of the service is to convert any type of content into a standard form that can be reused (RSS, XML). It also has the set of publishing features that turn content into Google Gadget, Netvibes Module, iCalendar, Flash [20] widgets and so on. It is a Web application that visually runs in a wizard mode asking the user to fill-in some field at every step in order to create a dapp (data imported). UI is very minimalist, but it gets the things done. Figure 8 shows dapper in action.

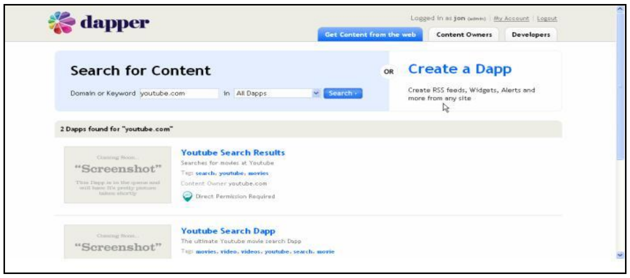


Figure 8. Dapper Data Mashup Maker

2.3.7 OpenKapow/Robo Maker

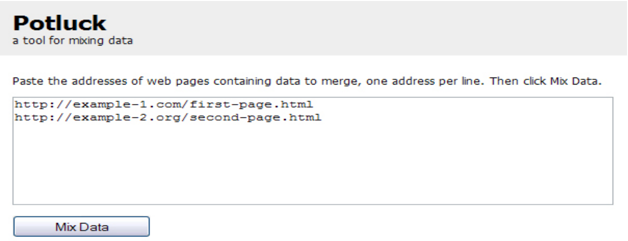
OpenKapow [13] is Web Mashup tool. It works with the concept of "robots" - you download a desktop application called Robomaker which is used to gather data from Websites. While Dapper is good at recognizing important data chunks on sites, Robomaker takes this to the next level, allowing you to automate complex processes and simulate a real person's behavior in a Web browser to retrieve the data you need. You can then create three different types of robots - RSS, REST or Web Clip robots, which enable you to either create RSS feeds, create an API out of a Web site or simply collect one piece of functionality from a site and use it somewhere else. All this makes OpenKapow a good tool for serious Mashup making (see Figure 9).

openkapow beta
Home
Robots
Downloads
Demos
Learn/Tutorials
Forums
Blog
Mashups in minutes
With openkapow you can quickly create, deploy, share and use web services (called Robots) that mashups anything on the web!
Learn more!
Internet
openkapow Robot
Services

Figure 9. OpenKapow

2.3.8 Potluck

Potluck [9] is a tool that lets end-users—non-programmers—make Mashups. Potluck allows the user to merge fields from different data sources, so that they are treated identically for sorting, filtering, and visualization. Fields are merged using simple drag and drop of field names.. Figures 10 and 11 show the interface of the Potluck.

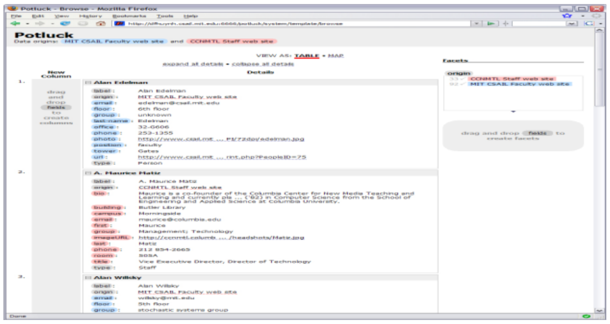


Potluck
a tool for mixing data

Paste the addresses of web pages containing data to merge, one address per line. Then click Mix Data.

```
http://example-1.com/first-page.html
http://example-2.org/second-page.html
```

Figure 10. The starting screen of Potluck takes URLs to Exhibit-powered Web pages. Clicking Mix Data yields the mixed data in a screen like in figure 15.



Potluck - Browse - Mozilla Firefox
File Edit View History Bookmarks Tools Help

Potluck
Data origins: MIT CSAIL Faculty web site and CCNMTL Staff web site

VIEW AS: TABLE • MAP
expand all details • collapse all details

New Column — drag and drop fields to create columns

Details

1. Alan Edelman
label: Alan Edelman
origin: MIT CSAIL Faculty web site
email: edelman@csail.mit.edu
floor: 6th floor
group: unknown
last-name: Edelman
office: 32-G606
phone: 253-1355
photo: http://www.csail.mit ... PI/72dpi/edelman.jpg
position: faculty
tower: Gates
url: http://www.csail.mit ... rint.php?PeopleID=75
type: Person

2. A. Maurice Matiz
label: A. Maurice Matiz
origin: CCNMTL Staff web site
bio: Maurice is a co-founder of the Columbia Center for New Media Teaching and Learning and currently pla ... ('82) in Computer Science from the School of Engineering and Applied Science at Columbia University.
building: Butler Library
campus: Morningside
email: maurice@columbia.edu
first: Maurice
group: Management; Technology
imageURL: http://ccnmtl.columb ... /headshots/Matiz.jpg
last: Matiz
phone: 212 854-2665
room: 505A
title: Vice Executive Director, Director of Technology
type: Staff

3. Alan Willsky
label: Alan Willsky
origin: MIT CSAIL Faculty web site
email: willsky@mit.edu
floor: 5th floor
group: stochastic systems group

Facets
origin
33 ✓ CCNMTL Staff web site
92 ✓ MIT CSAIL Faculty web site

drag and drop fields to create facets

Figure 11. Potluck's user interface shows data that has just been mixed together but not yet processed by the user. Fields are rendered as draggable "field tags," color-coded to indicate their origins. There are two drop target areas for creating columns and facets

2.3.9 Other General Purposes Mashup Makers

There are other Web Mashup creation tools which could be considered as general-purpose Mashup maker for end-users like Apatar [1], IBM Mashup center [10], Jackbe [11] and Serena [18].

2.4 Web Mashup Usability

Usability is a measure of the ease with which a system can be learned and used, its safety, effectiveness and efficiency, and attitude of its users towards it [15]. In general, Web usability refers to how well users can learn and use a Web site or a Web application to achieve their goals and how satisfied they are with that process.

It is important to realize that usability is not a single, one-dimensional property of a user interface. Usability is a combination of factors such as Ease of learning1, Efficiency of use2, Memorability3, Error frequency and severity4, Subjective satisfaction5. Usability should also be considered in the context of a specific group of users and taking into consideration the tasks that need to be accomplished.

Applying the concept of usability to Mashup makers, we can define Mashup usability as a measure of the ease with which a Mashup maker (tool) can be learned and used, its safety, effectiveness and efficiency, and attitude of its users towards it. In other word, Mashup usability is about user aspects and measurements when learning, creating, developing, building, using and handling Mashup creation tools.

Despite what is promised in general about Mashup makers, that it is very easy to develop Mashup and use those tools, we found that the Mashups general purposes development tools are still far away from "usable for end-users" and different efforts are still needed to enhancement and improvement of usability of Web Mashups.

2.5 Related work

There are many recent studies focusing on software usability impact factors and usability evaluation of software artefacts from various viewpoints [26]; [27]. However, little research is dedicated to the usability of Web Mashup Makers. In this sub-section we review some works related to usability of Web Mashup makers

2.5.1 Exploring Usability Guidelines for RIA

This work [8] is a master thesis of two graduate students at the department of informatics (The school of economics and management, The university of Lund in Sweden). They have used desktop usability guidelines and Web usability guidelines as a basis to create an outline of Rich Internet Application usability guidelines. Most of their work was focused on a comparative study of general usability guidelines. In conclusion they only formulated some so-called start guidelines for developer in the field of Rich Interactive Applications.

[1] ***Ease of learning*** *- How fast can a user who has never seen the user interface before learn it sufficiently well to accomplish basic tasks?[25]*

[2] ***Efficiency of use*** *- Once an experienced user has learned to use the system, how fast can he or she accomplish tasks?[25]*

[3] ***Memorability*** *- If a user has used the system before, can he or she remember enough to use it effectively the next time or does the user have to start over again learning everything?[25]*

[4] ***Error frequency and severity*** *- How often do users make errors while using the system, how serious are these errors, and how do users recover from these errors?[25]*

[5] ***Subjective satisfaction*** *- How much does the user* like *using the system?[25]*

Our research will focus on usability of Web Mashup makers and will be based on a usability study, experiments and usability testing.

2.5.2 MIT Potluck Usability evaluation

Potluck [9] is a project at the Computer Science and Artificial Intelligence Laboratory (MIT, USA). It aims at the development of an easy to use tool to Mashup data for end-users. They performed a usability evaluation study to ascertain whether people could learn how to use Potluck as well as to discover usability problems. Their study consisted of two tasks: a structured task and unstructured task. We have learned a lot from their experience and we will follow some of their notes related to the usability evaluation of Mashups tools.

2.5.3 Intel MashMaker Usability evaluation

Intel MashMaker [5] is a research project at Berkeley University (USA) funded by Intel. MashMaker is a Web-based tool to create Web Mashups by browsing around, without needing to type, or plan in advance what you want to do. Recently MashMaker is available as an extension to a Web browser such as Firefox. The research team of MashMaker has performed a usability evaluation of the tool following [27] and using the Cognitive Dimension of notations (CDs) framework [26]. That evaluation has helped us directing our intended study and experiments of usability of Web Mashups tools. Also their study of related work has helped us in selecting some Mashup development tools we should include in our research and experiments.

2.5.4 HUT, End User Mashup Programming Environments

An internal report by Oleg Beletski [2] at the HUT, Telecommunication Software and Multimedia laboratory (Helsinki University of technology, Finland) contains a study of some Web Mashup programming environment and compares usability basic aspects of those environments. The report has summarized the usability aspects of the compared Web Mashup programming environments (tools) by simply mentioning whether they are easy to use or not. The author has not mentioned any verification process for his measurements. Our research will deeply verify the usability of Web Mashup development tools.

2.5.5 Marmite usability evaluation

Marmite [21] is a research project of Human-Computer Interaction Institute (Carnegie Mellon University, USA). It is an end-user programming tool which lets users create Mashups that repurpose and combine existing Web content and services, Marmite is targeting users with programming backgrounds and with spreadsheeting skills. The development team of Marmite has performed a usability evaluation study which showed some difficulty for some users and the team intended to improve the usability aspects of the tool in the next versions. The Marmite usability evaluation study has helped us understanding a new way of evaluating Mashup development tools, which we will also consider in our usability study and experiments of Mashup development tools.

2.5.6 An end-user perspective of Mashup makers

In an internal report at the University of Victoria in Canada, the authors reviewed six Mashup makers from the so-called End User Development perspective. They mixed their reviewing methodology between some selected dimensions of the CD´s framework [3], software engineering techniques and some concepts related to e-learning. We have tried to fairly examine their report regarding to usability matters but we found it undetermined and in our opinion, it lakes specific usability review points.

Our literature research and first investigations have found that there is no complete and comprehensive work about Web Mashup usability at the time of writing this paper.

3. METHODOLOGY

One of the goals of the research work is to come to a suitable set of usability evaluation criteria for general-purposes Mashup makers for end-users. Among different usability evaluation methods that we have reviewed, we have selected the cognitive dimensions (CDs) of notations framework [3] [7] because of its nature as task-specific, and concentrating on the process and activities. The framework also targets visual programming tasks, which makes it very suitable for Mashup makers as these tools usually use a visual language. CDs framework provides a vocabulary that enumerates concepts important to variant users (skilled or not skilled) who are engaged in visual programming tasks. However, note that usability evaluation against cognitive dimensions is subjective, and it is not a substitute for thorough user evaluation. These concepts have been shown over time to be important to human problem solving and it is important to consider each when designing a usable artifact or interface [3] [7]. Table 2 shows a brief description of the cognitive dimensions and questions to be answered when evaluating usability of visual programming environments according to the CDs framework.

Table 2. CDs framework description [7]

Cognitive Dimension	Description
Abstraction Gradient	What are the minimum and maximum levels of abstraction? Can fragments be

	encapsulated?
Closeness of mapping	What 'programming games' need to be learned?
Consistency	When some of the language has been learnt, how much of the rest can be inferred?
Diffuseness	How many symbols or graphic entities are required to express a meaning?
Error-proneness	Does the design of the notation induce 'careless mistakes'?
Hard mental operations	Are there places where the user needs to resort to fingers or penciled annotation to keep track of what's happening?
Hidden dependencies	Is every dependency overtly indicated in both directions? Is the indication perceptual or only symbolic?
Premature commitment	Do programmers have to make decisions before they have the information they need?
Progressive evaluation	Can a partially-complete program be executed to obtain feedback on "How am I doing"?
Role-expressiveness	Can the reader see how each component of a program relates to the whole?
Secondary notation	Can programmers use layout, colour, or other cues to convey extra meaning, above and beyond the 'official' semantics of the language?
Viscosity	How much effort is required to perform a single change?
Visibility	Is every part of the code simultaneously visible (assuming a large enough display), or is it at least possible to juxtapose any two parts side-by-side at will? If the code is dispersed, is it at least possible to know in what order to read it?

We have conducted a study on eight general purpose Mashup tools: Yahoo Pipes, Microsoft Popfly, Intel Mashmaker, Openkapow Robomaker, Jackbe, IBM Mashup Center, Apatar and Dapper (Table 3).

Table 3. General-purpose Mashup makers considered

Mashup Maker	**Abbreviation**	**URL**
Yahoo Pipes	YP	http://pipes.yahoo.com
Microsoft Popfly	MP	http://www.popfly.com
Intel mashmaker	IM	http://mashmaker.intel.com/Web
Openkapow robomaker	OK	http://www.openkapow.com
IBM Mashup Center	IC	http://www.ibm.com/software/
Jackbe	JB	http://www.jackbe.com
Apatar	AP	http://www.apatar.com
Dapper	DA	http://www.dapper.com

We have conducted five main activities for each Mashup maker. The first activity is exploring the Mashup maker. In this activity we tried to discover the Mashup tool from an end-user perspective taking in consideration our target user (end-user). Exploring a Mashup tool includes discovering most of components and reviewing available learning materials. The second, third and fourth activities are the three main steps of any Mashup creation process: aggregating data, manipulate data and visualize data. We have tried to create similar Mashup with all of the eight tools. The fifth activity is a full Mashup creation example. For each Mashup tool and for each activity, we have given a qualitative evaluation for the different cognitive dimensions. We performed this study in a systematic way, which was quite time-intensive. For the qualitative evaluation, we ranked our observations of the usability evaluation by five levels. Code 5 is used for 'Very high level', code 4 for 'High level', code 3 for 'Moderate level', code 2 for 'Low level', and code 1 if we observe that the dimension was Missed in the tool within the activity under consideration. The ranking is summarized in Table 4.

Table 4. Ranking activity by dimension level

Dimension level	Very high	High	Moderate	Low	Missed
Rank	5	4	3	2	1

4. RESULTS AND DISCUSSION

The results of the study explained in section 3 are summarized in table 5. In this section, we clarify the results obtained from the study. Results ranking in tables 5 are calculated as an average of the summation of the ranking for the five activities performed .

Table 5. CDs Evaluation for Mashup makers considered

Cognitive Dimension/ Mashup Maker	**YP**	**MP**	**IM**	**OK**	**IC**	**JB**	**AP**	**DA**
Abstraction Gradient	4	3	4	5	3	4	4	3
Closeness of mapping	4	4	3	3	3	5	5	4
Consistency	4	3	4	4	3	4	4	3
Diffuseness	4	4	3	4	5	4	4	4
Error-proneness	2	1	3	2	3	2	2	1
Hard mental operations	2	2	2	2	2	3	3	2
Hidden depend-	3	4	3	4	2	4	4	4

encies								
Premature commitment	4	4	2	5	4	3	4	4
Progressive evaluation	1	2	3	3	2	3	3	2
Role-expressiveness	4	3	4	5	4	3	4	3
Secondary notation	2	3	3	3	2	2	3	3
Viscosity	4	4	3	5	3	2	2	4
Visibility	4	5	4	4	4	4	4	4

Furthermore we comment, discuss and evaluate the results for the different cognitive dimensions:

Abstraction Gradient: Mostly all the considered Mashup makers obtained an abstraction gradient level that is generally acceptable; some of them got a high level for this dimension. This allows us to conclude that most Mashup makers provide a minimal level of abstraction.

Closeness of mapping: The diversity of Mashup creation mechanisms provided by the different Mashup makers made it difficult to distinguish among the levels of required programming learning, but in general we have noted that almost all the Mashup makers provide tools with a simple approach and only some basic data flow understanding is needed. We think this dimension could be omitted in the case of usability test of end-user with no programming backgrounds.

Consistency: We noted that the Mashup makers provide a step-by-step Mashup creation process which makes it easy to users to infer the rest of the process.

Diffuseness: We noted that Mashup makers provide different shapes of symbols but in general we found them close to each other in providing an acceptable number of graphic entities to express meanings, which means there is a considerable level of diffuseness.

Error-proneness: We noted that almost all the Mashup makers considered in our study don't provide robust mechanisms for careless error handling.

Hard mental operations: For this dimension, we noted that Mashup makers lack facilities to support users to keep track of decisions or what is happening? So it may be useful to further investigate this dimension during the further usability tests for end-users.

Hidden dependencies: We noted that most of the Mashup makers considered in our study don't provide sufficient visibility of relations, and that's why it is clear that for this dimension the ranking is in generally high.

Premature commitment: The premature commitment is notably high by mostly all the inspected Mashup makers, we can justify this due to the nature of the Mashup makers as visual tools, this include commitments to layout, connections, order of creation, and choice of construction components. So users should make premature commitment mostly for every step during the Mashup creation process.

Progressive evaluation: Mainly, the inspected Mashup makers don't provide countable progressive evaluation. In general, the user should complete the whole creation process in order to get a tangible execution of the created Mashup. In our opinion, this functionality should preferably be provided to avoid that the user is continuing with a mistake or going in the wrong direction.

Role-expressiveness: We observed that the Mashup makers considered in our study provide sufficient role-expressiveness availability. Mainly there are relations and sub-relations between most of the Mashup makers components represented sometime by arrows and other times by successive layouts.

Secondary notation: We noted that most of Mashup makers considered and inspected in our study don't pay enough attention to providing secondary notation capabilities. Some of these secondary notations as commenting, indenting, naming conventions and grouping of related statements are available on a moderate level (ex. Intel Mashmaker). This may be a good point to focus on in further usability experiments that we will performed.

Viscosity: Here we noted that mostly all inspected Mashup maker don't provide the ability to make a single change without changing the whole Mashup. This is due to the nature of the Mashup, which is rather small and very specific. So we feel that the level of viscosity in Mashup makers is considerably high. At the same time we think that there will be no benefit of considering this dimension in the further usability test for end-users because there is in general no need to make changes in existing Mashups and providing this may also been too confusing for end-users.

Visibility: For this dimension, we noted that during our observations most of the Mashup makers provide interfaces with different components. We will examine this dimension further in our usability experiments to compare achievements and to get a better understanding of the evaluation of this dimension.

Discussion

We noticed that our study has several limitations.

1. First of all, as we did the evaluation ourselves, it was not always possible to correctly give a rank to the different dimensions, as we are not end-users. Therefore, it is important that we should redo the evaluation with members from the target users.
2. It may also be necessary to divide the target users further into groups based on their computer skills, e.g. their skills in Web surfing and handling Web browsers.
3. It may also be necessary to divide the target users further into groups based on their background in English, because all considered Mashup makers provide their interfaces in English.
4. To allow for a better comparison between the different tools in further experiments, a well-defined task/Mashup maker experimentation environment should be developed. This environment should include an example oriented towards the target user group, and should also provide learning materials and some know-how tutorials.

Because of the limitations of this study, the ranking provide for the different Mashup makers should not be considered as definitive. Despite this limitation, the study itself was very interested as we were able to detect which of the dimensions are useful to consider in further experiments (abstract gradient, consistency, diffuseness, error-proneness, hard mental operations, hidden dependencies, premature commitment, progressive evaluation, role expressiveness, secondary notation and visibility dimensions) and which should be omitted (closeness of mapping and viscosity dimension).

We also have to mention that some of the related works that we have studied and mentioned in our review have guided us in understanding our goal and steps required to reach our objectives of evaluating usability of general purposes Web Mashups makers.

5. CONCLUSION

Web Mashups combine information from multiple sources to produce a unified view of information to Web users. Currently, Web Mashups receive a lot of attention both from industry and researchers. Mashups promise to be the new way to "program" for the Web. Different tools for creating Mashups exist and some especially target novice or end-users, i.e. users with little or no background in programming. Using these Mashup makers it should be easy for them to quickly tailor and combine existing information for their own purpose. However, an important question is if indeed the available tools for Mashups are satisfying this promise, i.e. how high is their usability with respect to novice users. In this paper, we have reviewed some general-purpose Mashup makers; we have discussed a methodology to evaluate the usability of these Mashup makers and we have performed a study to evaluate the usability of 8 Web Mashup makers using this methodology.

5. REFERENCES

[1] Apatar, http://www.apatar.com

[2] Beletski, O., "End User Mashup Programming Environments", April 2008, available at: http://www.tml.tkk.fi/Opinnot/T111.5550/2008/End%20User%20Mashup%20Programming%20Environments_p.pdf

[3] Blackwell, A. F., Britton, C., Cox, A., Dautenhahn, K., Green, T. R. G., Gurr, C., Jones, S., Kadoda, G., Kutar, M. S., Loomes, M., Nehaniv, C. L., Petre, M., Roast, C., Roe, C., Russ, S., A., W., & Young, R. M., "Cognitive Dimensions of Notations: Design Tools for Cognitive Technology". In M. Benyon & C. L. Nehaniv & K. Dautenhahn (Eds.), *Cognitive Technology 2001* (pp. 325-341), Springer-Verlag.

[4] Dapper, http://www.dapper.net reviewed August 2014.

[5] Ennals, R., Gay, D., "User-Friendly Functional Programming for Web Mashups", ICFP'2007 available at: http://portal.acm.org/citation.cfm?id=1291187

[6] Google Mashup Editor, http://code.google.com

[7] Green, T., Petre, M., "Usability Analysis of Visual Programming Environments: a 'cognitive dimensions' framework", Journal of Visual Languages & Computing, Volume 7, Issue 2, June 1996, Pages 131-174.

[8] Gwardak, L., Påhlstorp, L., "Exploring Usability Guidelines for Rich Internet Applications", Master thesis Lund University, 2007, available at: http://biblioteket.ehl.lu.se/olle/papers/0002774.pdf

[9] Huynh, D., Miller, R., Karger, D., "Potluck: Data Mash-up Tool for End-users", ISWC'2007 available at: http://people.csail.mit.edu/dfhuynh/research/papers/iswc2007-potluck.pdf

[10] IBM Mashup center, http://services.alphaworks.ibm.com/graduated/Mashupcenter.html

[11] JackBe, http://www.jackbe.com

[12] Microsoft Popfly, http://www.popfly.com

[13] OpenKapow, http://openkapow.com/Default.aspx

[14] Ort, E., Brydon, S., and Basler, M., "Mashup Styles", reviewed August 2014, available at: http://java.sun.com/developer/technicalArticles/J2EE/Mashup_1/

[15] Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., & Carey, T., Human-computer interaction. Wokingham, UK: Addison-Wesley, 1994.

[16] Programmable Web,

http://www.programmableWeb.com reviewed September 2014.

[17] Simon Peyton Jones, "Wearing the hair shirt: a retrospective on Haskell" (invited talk). In ACM SIGPLAN Conference on Principles of Programming Languages (POPL'03), 2003.

[18] Serena, http://www.serena.com

[19] Wikipedia, http://en.wikipedia.org

[20] World Wide Web Consortium, http://www.w3c.org

[21] Wong, J., and Hong, J., I., "Making Mashups with marmite: towards end-user programming for the Web". In CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems, pages 1435 -1444, New York, NY, USA, 2007. ACM Press.

[22] Wong, J., Hong, J., "Patterns in Mashups", WEUSE IV 2008, ACM 1-58113-000-0/00/0004

[23] Yahoo Pipes,http://pipes.yahoo.com reviewed September 2014.

[24] Zillner, T., "Mashing It Up", A talk at Annual WAAL Conference , reviewed October 2014, available at: *http:*//www.wils.wisc.edu/events/waal07/Mashing_It_Up.Webbed.ppt

[25] http://www.usability.gov

[26] Ham, D.-H., Heo, J., Fossick, P., Wong, W., Park, S., Song, C., , Bradley, M., (2006). Conceptual Framework and Models for Identifying and Organizing Usability Impact Factors of Mobile Phones", Published by ACM OZCHI 2006, , Sydney, Australia.

[27] Seffah, A., Donyaee, et., al., (2006). Usability measurement and metrics: A consolidated model", Software Quality Journal, Vol. 14 (2), pp. 159-178.

Biographies

WAEL AL SARRAJ received the B.S. degree in Computer Engineering from the Islamic University of Gaza, Gaza, Palestine, in 2000, the M.S. degree in Electronic-Business Management from the University of Salento, Lecce, Italy, in 2002, and the Ph.D. degree in Computer Science from the Vrjie Universiteit Brussel, Brussels, Belgium 2012, respectively. Currently, He is an assistant Professor of Computer Science at the faculty of Information Technology at the Islamic University of Gaza. His teaching and research areas include Web engineering, Human Computer Interaction, and Software engineering. Dr. Wael Al Sarraj may be reached at wsarraj@iugaza.edu.ps