

## A Study of Mashup as a Software Application Development Technique with Examples from an End-User Programming Perspective

<sup>1,2</sup>Ahmed Patel, <sup>1</sup>Liu Na, <sup>1</sup>Rodziah Latih, <sup>2</sup>Christopher Wills,  
<sup>1</sup>Zarina Shukur and <sup>1</sup>Rabia Mulla

<sup>1</sup>School of Computer Science, Faculty of Information Science and Technology,  
University Kebangsaan Malaysia, 43600 UKM Bangi, Selangor Darul Ehsan, Malaysia

<sup>2</sup>Faculty of Computing Information Systems and Mathematics, Kingston University,  
Penrhyn Road, Kingston Upon Thames KT1 2EE, United Kingdom

---

**Abstract:** The purpose of this study is to present, introduce and explain the principles, concepts and techniques of mashups through an analysis of mashup tools from End-user Development (EuD) software engineering perspectives, since it is a new programming paradigm. **Problem statement:** Although mashup tools supporting the creation of mashups rely heavily on data integration, they still require users to have reasonable programming skills, rather than simply enabling the integration of content in a template approach. Mashup tools also have their lifespan in a fast moving technology-driven world which requires meta-application handling. Some developers have discontinued their mashup tools but others are still available in the mashup space. It has been noted that there is a steady increase of new mashups on a daily basis with a concomitant increase of new Application Programming Interface (APIs) to support meta-mashup application EuD. **Approach:** Both qualitative and quantitative research methods have been utilized. After introducing the basic principles, concepts and techniques of mashups, we develop and present a categorization of mashups and mashup tools and summarize the ten most popular currently used mashup tools against seven indicators from end-user software engineering perspectives ranging from programming skill requirement, prompt suggestion of features use, operability, 'share-ability' and reuse, service, type and target user, in order to evaluate how these mashup tools support end-user development. To perform the evaluation and produce the final results, the selected indicators' features have been horizontally compared and comprehensively analyzed. **Results:** The philosophy of mashup is aimed at providing simple rapid program development by end-users with minimum programming skills. However, we observe that mashup tools typically follow four data processing styles: Webpage customization, wire paradigm, spreadsheet and programming by demonstration. These mashup tools are supposed to seamlessly and effortlessly assist end-users programming but this is not the case. **Conclusion:** From our research we concluded that some mashup tools are not really simple enough to handle and still require end-users to have a computer programming background to learn and understand its platform infrastructures and mechanisms. These might all change in the near future. There are some companies that are now involved with mashup development which provide huge opportunities to both individual and organizational customers. Mashup have now become a commercial opportunity rather than a simple way of integrating data from Web 2.0 platforms.

**Keywords:** Mashup, mashup tools, meta-applications, software development, Webpage customization, Application programming interface

---

### INTRODUCTION

Mashup is a relatively new and emerging form of Web application development that is gaining widespread recognition. The original definition of mashups was derived from mixing music songs to create a new "remix" regardless of the sources as shown in Fig. 1. The

computing fraternity latched onto the word "mashup" by combing multiple services into a single high-order Web application (Hartmann *et al.*, 2007; Zang *et al.*, 2008).

A very recent mashup example was to help end-users to follow the BP oil spill crisis as a news item. An explosion on an offshore oil rig in the Gulf of Mexico on 20th April 2010 started a leak which spewed millions of

---

**Corresponding Author:** Ahmed Patel, School of Computer Science, Faculty of Information Science and Technology,  
University Kebangsaan Malaysia, 43600 UKM Bangi, Selangor Darul Ehsan, Malaysia

gallons of oil into the ocean. Mashup developers created mashups to understand and digest the massive amounts of news available. One example of such a site, using Google Earth APIs, called BP Gulf Oil Spill in the Gulf of Mexico affecting all coastal areas, which supported end-users to look around the approximate location of the latest observed spill vicinity via underwater photo view images as is shown in Fig. 2.

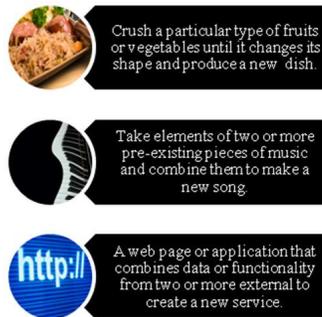


Fig. 1: Mashup with different expressions of remix



Fig. 2: BP gulf oil spill in the Gulf of Mexico interface

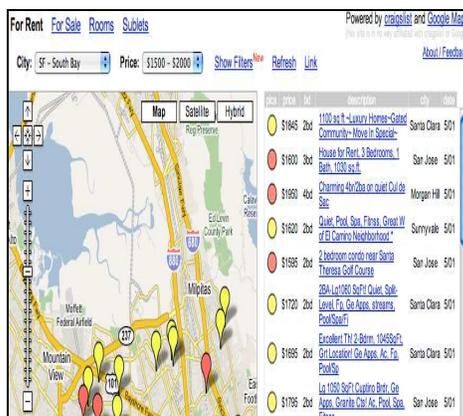


Fig. 3: The remix interface of housingmaps.com

Another example, housingmaps.com, as shown in Fig. 3, is a typical successful mashup. It was built by Paul Rademacher, an independent programmer. It is a mashup that combined property listings from Craigslist with map data from Google Maps to show not only the list of properties available, but also the respective locations and other property information. The mashup took place on the server of housingmap.com which was independent from its original sources: Craigslist and Google Maps. By integrating these popular Website resource services, housingmaps.com created a new fresh Web application.

However, the development of users' applications in the complex world of computing by following the pattern of writing "knit and stitch" programming is a daunting task. Creating a mashup also requires skills in programming language (Zang *et al.*, 2008) such as HTML, JavaScript, PHP and knowledge of relevant APIs which limits the ability of non-programmers to create mashups. Since 2006, several commercial mashup tools which focused on end-users, Web developers and business organizations have created such tools as Microsoft Popfly, Yahoo! Pipes and IBM Mashup Centre. These mashup development tools were created largely to focus on correlating data rather than writing complex mashup programs. Since then, both mashup research and tools developed rapidly and many new features were invented and added. Mashups have become simpler, user friendly, easy to use and access (Yu *et al.*, 2008).

## MATERIALS AND METHODS

**An overview of mashup applications:** Mashup refers to Website or Web applications which seamlessly integrate content from multiple sources into combinations of information patterns. Some examples of mashup applications where data and services are remixed to create new valuable application patterns are:

- Google Maps in Flickr (GMiF) integrate Flickr, Google Maps, Google Earth and Firefox together using Greasemonkey script
- Library Lookup bookmark based on JavaScript is a bookmark gadget that enables Amazon.com to integrate with your local library menu
- Yahoo! Pipes enables a user to synthesize a piece of more informative news feed from multiple news sources
- Google Calendar has dedicated complex APIs that enables adding Web events into one's personal calendar

**A simple mashup application example:** Flickr is a social network Website with huge flexibility that can be highly mixed. It provides not only basic photo storing and sharing functions, but, most importantly, it supports XML, XML Web services, Tag mechanism and AJAX. Figure 4 illustrates a mashup application in Flickr that allows end-users to display their photos in Google Maps platforms, by using Google Maps in Flickr (GMiF) which is a script implemented through a Firefox extension. GMiF works as a Greasemonkey script that rewrites HTML code in Flickr in order to insert a GMiF button into a photo canvas. In addition to the embedded Google Maps function, several JavaScript codes will be added when users click the GMiF button.

**Typical mashup pattern:** Normally, the creation of a mashup focuses on two aspects: firstly, to obtain content from different Websites and then apply the extracted contents as a combination of elements into a new application. This includes three basic operations: data extraction, data matching and data integration (Murth *et al.*, 2006). Data extraction is the process of obtaining data from source sites. Data matching is the prior transformation of that data into a format expected by the receiving terminal sites for assembly or integration into the desired application without further transformation.



Fig. 4: A mashup application with GMiF

**Most popular APIs and mashup tags:** We observed from ProgrammableWeb.com statistics (Table 1) that there were 4884 tracked mashups involving 2066 APIs on 31st July 2010. During the last 6 months, mashups increased from 4600-4950 with approximately 2 new mashups created or added daily and 4 new APIs added per week, giving approximately 23 new APIs per month with an average 0.7% rate of increase. Among these 2066 APIs, the top three popular ones were Twilio with 36%, TwilioSMS with 13% and with a shared tie position between Google Maps and Twitter at 12% each. But realistically speaking, it appears that GoogleMaps has an edge over Twitter since it provided the earliest “open” (publicly available and accessible) set of APIs for creating mashup applications and they are still the most popular in their widespread usage because of their early entry, exposure and utility or usefulness.

With over 150,000 sites already using the Google Maps APIs from the time of its launch, there are approximately 35,400 registrants per day and the registration rate is as high as 54%. And from Table 1 it can be further observed that from all tagged APIs analyzed by ProgrammableWeb.com, Mapping at 10% has fallen from when it was at the top most position at 20% when the Website was visited two months ago.

**Three common mashup types:** Categorizing mashup applications is generally perceived in the form of a multi-view in three different ways, namely, from its architecture, combined items and involved users from different usage perspectives. In these types of mashup, the most common functional elements are data, consumer and business mashups as shown in Table 2.

**Mashup techniques:** The Web is known as the largest and the most successful database and exchange system ever built. The creation of a Web application involves data processing that conventionally goes through three tiers of interconnected processing as shown in Fig. 5: tier-1 presentation, tier-2 application and tier-3 information storage (Petersen, 2009).

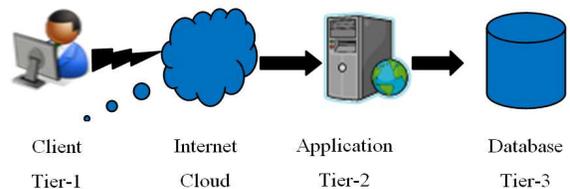


Fig. 5: Three tiers of Web meta-application creation

Table 1: Most popular APIs and mashup tags of various application types

Social	Twitter	
Mapping	Facebook Google Maps Microsoft Virtual Earth	<p><b>Top API for Mashups</b></p> <ul style="list-style-type: none"> <li>Twilio (36%)</li> <li>TwilioSMS (13%)</li> <li>Twitter (12%)</li> <li>GoogleMaps (12%)</li> <li>Facebook (6%)</li> <li>YouTube (5%)</li> <li>Flickr (3%)</li> <li>Foursquare (3%)</li> <li>MaxMindGeoIP (3%)</li> <li>GoogleAppEngine (3%)</li> </ul> <p>ProgrammableWeb.com 07/31/10</p>
Music	Last.fm Lyricsfly	
Shopping	Amazon eCommerce	
Enterprise	Salesforce.com	<p><b>Top Mashup tags</b></p> <ul style="list-style-type: none"> <li>telephony (22%)</li> <li>social (13%)</li> <li>messaging (12%)</li> <li>Voice (11%)</li> <li>sms (11%)</li> <li>mapping (10%)</li> <li>Twitter (6%)</li> <li>sports (5%)</li> <li>video (3%)</li> <li>Internet (3%)</li> </ul> <p>ProgrammableWeb.com 07/31/10</p>
Video	EchoSign YouTube	
Photo	AOL Video Flickr Panoramio	

Table 2: Mashups multi-view categorization

Category	Sub-category	Remarks
Architecture	Client mashup	Data is combined and reformatted in user's Web browser
	Server mashup	Data is analyzed and reformatted on a remote server
Combined items	Presentation mashup	Information is either remixed or just placed next to each other
	Data mashup	Similar types of media and information from multiple sources are combined into a single data source
Involved users	Process mashup	Functionality is combined together in one or more external processes using programming languages
	Consumer mashup	Visual elements and data is combined from multiple sources
	Business mashup	Aims to solve business problems

A Web browser running on an end-user's machine is the first tier that provides data display, permitting end-users to view, enter and update data. The application tier is the powerhouse which generates Webpages and it includes dynamic Web content technology such as ASP and JSP to facilitate this function. The database is the third tier of the Web application process which manages data hosting of content and supporting tag information that is typically organized in relational form (Liu *et al.*, 2007). Generally, a client sends requests to the Application tier, which is a Web server engine that connects to multimedia information Database (s) for searching and retrieving content, querying and updating and hosting new data content.

Web 1.0 refers to static HTML for publishing and downloading of Webpage content as specified by W3C norms. Compared with Web 1.0, Web 2.0 refers to social networking sites like Flickr that allow many people to build communities and upload and share content they have created in an interactive and collaborative manner

as shown in Fig. 6 and Fig. 7 respectively. On Web 2.0 platforms, users are not only single readers when they are surfing the Internet to search information sources and retrieve data, but they also help create content by blogging personal and personalized information easily and seamlessly through the functionality offered by the Web 2.0 platform and multimedia applications.

Since both the amount and diversity of Web and other content is expanding rapidly, a large number of resources are shared or shareable, integrated and republished among end-users, which encourage existing mashups to evolve and new ones to emerge as high-order Web applications that mushroom by recombining information from a bewildering array of data resources and sources. These high-order applications are classified as meta-applications.

In Web 2.0 platform, data is an independent item and can be aggregated and transferred easily by XML and its clones. Compared with HTML, XML simplifies the processing of retrieved data. This is because XML

emphasizes data transportation and storing, while HTML is concerned with displaying information. So the data retrieved from HTML require reprocessing to extract the information content from it. This information, turned into data input, can then be used for further remixing.

Moreover, Asynchronous JavaScript and XML (AJAX) is a group of interrelated Web development methods used on the client-side that supports integrating disparate content or services into a single user experience by dynamic access on any XML-based data source. With AJAX, data retrieval from the server is in the asynchronous context, without interfering with the display and behavior of the existing page.

On the data sharing side, the content used in mashups is typically sourced from a third party via open and publicly available interfaces, called APIs that are a set of functions pertaining to messages and data structures. In its simplest form, a Web API is actually a synonym for Web service that supports mashups in terms of allowing the combination of multiple services into new meta-applications. For example, Really Simple Syndication (RSS) plays an essential role in mashup publishing. It is an extension of information online sharing model by supporting filtering information to end-users via subscribing their most interesting information such as news, blogs or videos in real-time and publishes them as a family of multi-format Web feeds.

Mashup is considered as a new type of application where resources are loosely aggregated together to create a single high order meta-application with richer functionality. The appearance of more mashups is expected as Web 2.0 technology advances and matures, providing more opportunities for end-users to participate in new Web application development. To create a mashup application, there are two primary options; use APIs if you are familiar with it, or mashups tools if you are a novice without any programming skills. The end-user will select one or the other that is most comfortable and convenient to create his/her personal unique mashup. Selecting a suitable method to create their mashups is quite significant because it gives the personalized flavor to the mashup based on their likes and dislikes. We believe that the tools option is much easier, more efficient and will have a multiplier effect of success in the long run for the end-user.

**Mashup tools:** Mashup tools were developed to support non-programmers to create their own variety of mashups fulfilling different objectives and needs (Simmen *et al.*, 2008). Conventional mashups were created by the end-users who are professional at

programming language with particular information technology. With Web 2.0 platform, information sharing tends to be widely mature since the end-users' role has changed virtually from single internet resources' reader to an interactive writer in different shared community activity modules among end-users which has recently become more effective and frequent. The mashup tools aim to provide opportunities to every end-user that participates in customizing Web applications with personal interests and requirements.

Mashup tools developed by different owners provide distinct services as shown in Table 3. Microsoft terminated Popfly, (once a popular mashup tool), ceased Popfly operation and shutdown all related resources, supporting sites and links on 24th August 2009 because it is believed that the competition was eating into their budgets without viable financial returns, which Microsoft called "budget cutbacks".

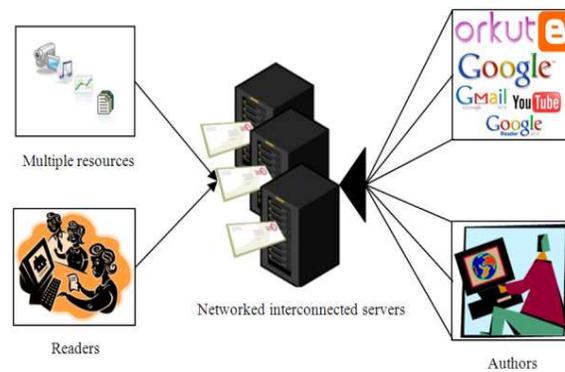


Fig. 6: Author and reader interaction and collaboration

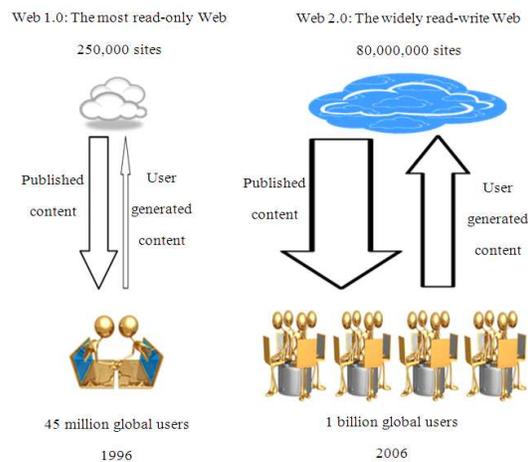


Fig. 7: End-users' role has changed from Reader-only to Editor between Web 1.0 to Web 2.0 generation

Table 3: The basic information of mashup makers

Mashup maker	Owner	Launched
Yahoo! Pipes	Yahoo!	February 7, 2007
iGoogle	Google	May 2005. Renamed as iGoogle on April 30, 2007
Microsoft Popfly	Microsoft	May 18, 2007. Discontinued on August 24, 2009
Apatar	Apatar	Project initiated in 2005. Stable version released on September 21, 2007
IBM Lotus Mashups (Formerly QEDWiki)	IBM	Showcased an early version on January 23, 2008
Google Mashup Editor	Google	Migrated to Google App Engine on January 14, 2009
Intel Mash Maker®	Intel	Released the mature 2nd beta version on April 22, 2008
Marmite		2007
Vegemite	IBM	2008
Dapper	Dapper	2005

However, Microsoft Popfly was regarded with skepticism due to its reliance on Microsoft's Silverlight Web platform, which focused on software rather than about data mashing (Ferrate, 2009). This is not unique; Google Mashup Editor that was once an AJAX development framework with a set of tools which enabled developers to quickly and easily create simple Web applications and mashups with Google services was shut down by Google on 14th January, 2009. It is disheartening to see these mashup tools in the development graveyard. It was quite difficult to measure which factors eventually determined the failure of their discontinuations. Fortunately, there are still some tools available including Yahoo! Pipes and Intel Mash Maker®. Mashups ecosystem is like any other systems which obey the survival of the fittest principle to maintain lifespan. We believe that in this fierce competitive, opportunistic and challenging arena mashups can coexist. The demised mashups will simply bring more opportunities to others in this space.

**Mashup tools category:** Mashup tools can be categorized through different data processing modules: Webpage customization, wire paradigm, spreadsheet and programming by demonstration.

Webpage Customization allows end-users to browse and design Webpages that they are going to integrate as a set. Changes to content take place when mashup developers search Websites, for example:

- iGoogle is a Web service based on Google gadgets that enables end-users to establish personalized homepages which contain a Google search box at the top. The homepages can be customized by adding Web feeds and Google gadgets that are displayed on a canvas. Many themes are provided to satisfy groups of end-users with different flavors. These features include Google bookmarks, weather, Google search and also Wikipedia search can be integrated and displayed in iGoogle page
- Intel Mash Maker® is an interactive tool that helps users to track the current tasks which they are

doing and provide information and visualizations about what might be useful for their tasks (Ennals and Garofalakis, 2007). It is plugged in, worked and played with the end-users browsers. The contents are from multiple resources such as Websites, videos, maps, RSS feeds, tweeds and photos which are published in one place. Mash Maker® provides tools that allow end-users to manage their mashups by copy-paste, editing, sorting and annotating

**Wire paradigm:** Wiring paradigm provides a series of services in the forms of modules, connectors, components or blocks that are called pipes or wires. Developers can choose the suitable pipes or wires that support particular functions they want and drag and drop them onto a canvas. For example:

- Yahoo! Pipes is a powerful composition tool to aggregate, manipulate and combine content from the Web (Lin *et al.*, 2009). Yahoo! Pipes supports end-users to achieve their requirements by combining, filtering and translating custom feeds that address the user's specific objectives and needs by providing powerful widgets or gadgets. Yahoo! Pipes enables end-users to search, share and reuse pipes with other users. The terminal - end-user side - set of pipes are published as RSS, JSON, KML and other formats for ease of use
- Apatar is a desktop application installed at the client side to provide end-users with connectors, data services and operators to integrate information on the basis of on-premise or on-demand for the data and applications. Applications are connected on the spot for non-developers with graphic tools and data transformers with visual mapping and aiding tools and without coding requirements. ETL (Extract-Transform-Load) engine enables end-users to provide their applications in the way they want to their customers or business partners to use. Apatar extends its support by providing customers with consulting, data integration and training services

Table 4: Features of the mashup tools

Name	Programming skill requirement	Prompt suggestion of features to use	Operability	Share-ability and reuse	Service	Type	Target user
Yahoo! Pipes	Average	No	Average	Yes	Hosted service, Visual programming environment	Remixing develop	Individual
iGoogle	Non-technical	Yes	Low	Yes	Customizable start page Personal Web portal	Web gadgets	Individual
Apatar	Average	No	High	Yes	Data integration Software application	ETL data migration to SQL	Organization
IBM Lotus Mashups	Non-technical	Yes	Average	Yes	Browser-based tools widgets	Enterprise and Web data platform	Organization
Intel Mash Maker®	Expert non-technical	No	Average	Yes	Toolbar	Extension client to Firefox Web browser	Individual
Marmite	Average	Yes	Low	Yes	Toolbar	Extension client to Firefox Web browser	Individual
Vegemite	Average	No	Average	Yes	Hybrid data-flow Incremental programming paradigms	End-user programming system	Individual
Dapper	Average	Yes	Average	Yes	Virtual browser Nontechnical interface Widgets, create APIs	Structured information platform	Organization

- IBM Lotus Mashups is a mashup tool running in a browser based on a wiki concept called QEDWiki, where QED stands for *quick and easily done*. QEDWiki provides mashing up different resources to create end-user desired applications quickly to meet instant needs (Elmeleegy *et al.*, 2008). It provides various supporting tool functions to assist editing, commenting, publishing, emailing and collaborating. QEDWiki runs in a browser based on Web 2.0 technology like AJAX creating rich end-users' interactive experiences. Lotus Mashups is considered as a framework of QEDWiki that was launched in the annual IBM Lotusphere conference in 2008. It aims at rapidly assembling mashup applications using browser-based visual tools for end-users and it also provides opportunities for adding more powerful tools to support end-users to create sophisticated mashups for business domains. QEDWiki and IBM Lotus Mashups share the same technology from IBM's Emerging Technology Group but Lotus Mashups is a separate commercial product from QEDWiki.

**Spreadsheet oriented:** In a spreadsheet-oriented mashup framework the integrated data is directly inserted into a spreadsheet. It is an easy mashup platform where the advantages are to enable end-users to see the current state of the processed data and help to view the bugs dynamically while the processing takes place with the possibility of allowing real-time online debugging. For example:

- Marmite works by integrating data with screen scrape-oriented programming and then displaying the integrated data as flow or spreadsheet views. The programming progress as well as the data can be seen simultaneously (Wong and Hong, 2006; 2007) as the mashup is created

**Programming by demonstration:** Whereby end-users are able to learn new operations through the provision of examples in template form instead of programming-like style. This method of creating mashup is exemplified by such tools as:

- Vegemite that uses direct manipulation and programming-by-demonstration techniques to automatically populate tables with information
- collected from various Websites (Lin *et al.*, 2009)
- Dapper that uses screen-scrapers to allows end-users to access any Webpage and then select any one of the output format template to extract data

**Mashup tools features:** We selected seven feature themes with the aim of exploring end-users development aspects, especially for non-programmers in order to evaluate their usability, operability and user-friendly interface as summarized and shown in Table 4, with the following proviso:

- Programming skill requirement: To create a mashup. Normally, it is expected for end-users to have some programming skills. This can be determined from three distinct groups of end-users with different programming backgrounds: non-technical, average and expert. It can also be used for evaluation purposes to analyze and determine skill levels, output productivity and use of mashup facilities
- Prompt Suggestion of Features to Use: Refers to the embedded system mashup directional facility which offers relevant hints to assist end-users follow and understand their meta-application creation activities in real-time
- Operability: Refers to the level of abstractions and ease of learning for end-users

- Share-ability and Reuse: Refers to the developed mashups that could be shared and reused as a meta-application with other mashups and developers
- Service: Includes widgets, toolbars, visual interfaces and paradigms that mashup tools use to simplify the progress of creating mashups
- Type: Refers to the platform mashup tool's reliance on a browser, remixing data or end-user programming system
- Target user: Refers to the target group of end-users for whom these mashup tools have been developed for: individual or organization

## **RESULTS AND DISCUSSION**

Over half of the reviewed mashup tools in this study need their end-users to partly have average programming skills. Only iGoogle and IBM Lotus Mashups support non-technical end-users with widgets, but the function does not fully satisfy end-users' needs. Intel Mash Maker® is considered as the tool for both (i) non-technical users to manipulate content through a set of functions and (ii) experts using it explicitly to specify a formula type advanced functions and relations into which data is fed by mashup maker flow dynamically to create content (Ennals and Garofalakis, 2007).

Yahoo! Pipes is a remixing development platform, which allows meta-applications from different functions and operations to be assembled and integrated by connecting pipes through a set of operators in a programming-like manner. iGoogle and Mash Maker® is a Webpage-oriented platform, where data processing takes place simultaneously with adding widgets to the current edited Webpages. Marmite is an extension plugged in the Firefox browser which assists in creating mashup more easily. Vegemite implements a data-flow programming paradigm, while dapper focuses on structured information system that concentrates on data extraction and integration.

Apatar does not only provide client application software, but it also provides other services to their customers, such as data integration, ETL migration to SQL, customization, training and consulting services. Dapper is a dynamic tool displaying advertising technology which is based on a different mashup application model to help vendors to find those customers who closely match the selling or buying patterns or profile of their product lines.

All the mashup tools which were reviewed in this study support help-like facilities and automatic prompting of suggestive features to give relevant hint

and direction to end-users in creating their meta-application. The majority of them provide a startup button or icons to get started easily and at each stage they may pop prompt boxes to assist the user to follow the most likely next step.

Yahoo! Pipes allows sharing of pipes with other mashup developers while the end-users pipes can be found by other developers who are interested in that subject/topic area. Mash Maker® supports end-users who wish to share their mashups with their friends or with social networking community sites. Using this tool, end-users can easily design and tailor their mashups through the visualization features and functions to customize control the layout and view their mashups. Dapper enables end-users to search RSS feed that have been published by other developers and build them up as a meta-application.

The majority of targeted end-users of the mashup tools that were reviewed are individuals who are interested in establishing their Web application as a meta-application. Only IBM Lotus Mashups is one that aims to provide enough powerful tools for the business domain. It is considered as a commercial application product based on IBM's Mashups Centre.

In addition to those big corporations, many small and medium sized companies have also been involved in the development of mashup tools and facilities. Some of them have achieved huge success such as Apatar whose headquarters is now in Massachusetts and currently used by 3500 organizations and individuals worldwide for creating mashups.

From the tools assessed by this study, iGoogle is the simplest and easiest mashup tool for novice end-users. It does not require any programming or specific background knowledge so that end-users can create their own cookbook applications within a few minutes. However, all of the reviewed mashup tools need end-users to learn how to use them before starting to create mashup applications. On the other hand Yahoo! Pipes requires an average level of understanding of all the functions and operations of each connector and operator. Apatar also requires a deeper level of understanding of abstraction by end-users of things such as database knowledge, in order to abstract and connect data from different diverse data sources.

We selected, categorized and reviewed ten mashup tools to display their features. We found that such tools provide different services and adapt to different target end-users or customers. iGoogle and Mash Maker® based on Web customization provide widgets and gadgets that enable end-users to edit target Webpages to retrieve and integrate data. Starting off with this kind of tool is easy and rapid progress can be made without

much outside support, thus individuals are the main customers of their use. Yahoo! Pipes, Apatar and IBM Lotus Mashups are based on the wire paradigm concept for selecting connectors and visually wiring them onto the canvas for subsequent processing and display that makes the whole creation exercise of a meta-application easy to handle. However, before using this type of tool, the end-user must learn something of the features it provides, define the requirements and understand the functionality of the meta-application in order to select the appropriate connectors to be wired. However, for non-technical developers or novice end-users, the learning process might be long and complex, even after being introduced to how to create mashups because there is no better experience than what comes from practice. This is a necessity that is hard to pass by. What has to be learnt must be learned and practice makes it easy!

Marmite is a mashup which uses spreadsheet as a technique that readily stores and displays complex data in a variety of relational forms. The retrieved data is displayed as one would find in a typical spreadsheet program, such that it is organized neatly and the process is governed by a set of logical rules and formulae-type operations. However, graphics or images can be easily added to enhance the meta-application output presentation display.

In programming by demonstration/example, end-users instruct the system in what they want to do with the data and the system will then complete the task automatically through interpretative rules. While the system is running, end-users cannot interrupt the process until it has run to completion. It has to be performed as an atomic action because of the complexity of the data sets and multi- or cross-platform processing cycles. This method of mashup creation can be time-consuming and tedious for end-users if the data item is large and the process is computationally complex, intensive and repetitive.

An important requirement for mashup meta-applications is their authenticity and integrity. Although outside the scope of this study, the problem with mashups is that the content or information can be abused through various forms of cyber vandalism, commonly known as trolling which can go undetected during the live time existence of the mashup. Their processing can be equally hijacked and misused to present covert data displays or diverted to other mashup sites. Mashups need security in the form of a set of safety measures, regardless if they operate as fixed or mobile software agents. The safety measure (Patel, 2010) has to comprise of security, privacy, trust, auditing and digital forensic functions and their

enforcement will depend on the confidentiality and sensitivity of the mashup and its data content.

## CONCLUSION

In this study of mashup as a software application development EuD technique, we presented an emerging phenomenon and trend offered primarily by Web 2.0 technology. Web 2.0 has boosted mashup making and content sharing by the ordinary “non-programmer type” community of users through the use of relatively simple and easy to use instructions, functions and operations. They require minimum or no technical support, due to the use of “open” APIs, AJAX and RSS as a group of interrelated Web development methods. They offer individuals the opportunity to create, upload and share personalized remix content through diverse data integration from different sources, giving end-users unique experiences in mashup building and display with the ultimate goal of sharing content with friends and foes over the Internet at rapid speed and minimum cost.

From the review it was observed that the number of mashup applications is increasing since mashup development tools have become available to support end-user meta-application development. The majority of current mashup tools support end-user programming but to create a mashup meta-application successfully, end-users still need to have some programming background and knowledge. However, this has not hampered mashup development as a technique or restricted their use. Mashup frameworks and architectures are now transforming from simple content or data integration to multi-faceted meta-applications that have valuable commercial implications and offerings other than just end-user social networking.

We believe that on the one hand mashup really brings more opportunities with a new revolution to the majority of on-demand creative end-users while on the other hand challenges still exist in some mashup areas such as technology maturity in the business domain to make them a common practice. The latter is our next study to find out what will be the next generation of mashups in the business domain that will influence new development paradigms, methods and processes.

## REFERENCES

- Elmeleegy, H., A. Ivan, R. Akkiraju and R. Goodwin, 2008. Mashup advisor: A recommendation tool for Mashup development. Proceeding of the Web IEEE International Conference on Services, Sept. 23-26, IEEE Xplore Press, Beijing, pp: 337-344. DOI: 10.1109/ICWS.2008.128

- Ennals, R.J. and M.N. Garofalakis, 2007. MashMaker: Mashups for the masses. Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, June 11-14, ACM Press, Beijing, China, pp: 1116-1118. DOI: 10.1145/1247480.1247626
- Ferrate, A., 2009. Microsoft shuts-down its popfly mashup tool. Programmableweb. Com. <http://blog.programmableweb.com/2009/07/16/microsoft-shuts-down-its-popfly-mashup-tool/>
- Hartmann, B., L. Wu, K. Collins and S.R. Klemmer, 2007. Programming by a sample: Rapidly creating Web applications with d.mix. Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology, Oct. 7-10, ACM Press, Newport, Rhode Island, USA., pp: 241-250. DOI: 10.1145/1294211.1294254
- Lin, J., J. Wong, J. Nichols, A. Cypher and T.A. Lau, 2009. End-user programming of mashups with vegemite. Proceedings of the 13th International Conference on Intelligent User Interfaces, Feb. 8-11, ACM Press, Sanibel Island, Florida, USA., pp: 97-106. DOI: 10.1145/1502650.1502667
- Liu, X., Y. Hui, W. Sun and H. Liang, 2007. Towards service composition based on mashup. Proceedings of the IEEE Congress on Services, July 9-13, IEEE Xplore Press, Salt Lake City, UT., pp: 332-339. DOI: 10.1109/SERVICES.2007.67
- Murth, S., D. Maier and L. Delcambre, 2006. Mashomatic. Proceedings of the 2006 ACM Symposium on Document Engineering, Oct. 10-13, ACM Press, Amsterdam, The Netherlands, pp: 205-214. DOI: 10.1145/1166160.1166214
- Patel, A., 2010. Concept of Mobile Agent-based Electronic Marketplace-Safety Measures. In: Encyclopedia of E-Business Development and Management in the Digital Economy, Lee, I. (Ed.). IGI Global Snippet, Macomb, IL., USA., ISBN: 1615206116, pp: 252-264.
- Petersen, J., 2009. Benefits of using the n-tiered approach for web applications. Adobe.Com.
- Simmen, D.E., M. Altinel, V. Markl, S. Padmanabhan and A. Singh, 2008. Damia: Data mashups for intranet applications. Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, June 9-12, ACM Press, Vancouver, Canada, pp: 1171-1182. DOI: 10.1145/1376616.1376734
- Wong, J. and J. Hong, 2006. Marmite: End-user programming for the web. Proceedings of the CHI '06 on Human Factors in Computing Systems, April 22-27, ACM Press, Montréal, Québec, Canada, pp. 1541-1546. DOI: 10.1145/1125451.1125733
- Wong, J. and J.I. Hong, 2007. Making mashups with marmite: Towards end-user programming for the web. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Apr. 28-May 3, ACM Press, San Jose, California, USA., pp: 1435-1444. DOI: 10.1145/1240624.1240842
- Yu, J., B. Benatallah, F. Casati and F. Daniel, 2008. Understanding mashup development. IEEE Internet Comput., 12: 44-52. DOI:10.1109/MIC.2008.114
- Zang, N., M.B. Rosson and V. Nasser, 2008. Mashups: Who? What? Why? Proceedings of the CHI'08 extended abstracts on Human Factors in Computing Systems, Apr. 5-10, ACM Press, Florence, Italy, pp: 3171-3176. DOI: 10.1145/1358628.1358826