USING MASH-UPS IN DEVELOPMENT OF INTGERATED DATA PRESENTATION APPLICATIONS

Matins Bonders, Andrejs Abele

Riga Technical University, Institute of Information Technology,
1 Kalku Street, Riga, Latvia, martins.bonders@rtu.lv, andrejs@iti.rtu.lv

Abstract. Mash-ups are one of the application integration technologies. They are aimed at reducing integration complexity and are advocated as one of the most promising technologies for user-driven integration. Mash-ups allow consumers to draw upon content retrieved from external data sources (web services, data store and web application) and to create new services. Mash-ups use general Web browsing principles and have user interface while web service are primarily designed to support machine-to-machine interaction over a network. Availability of the presentation layer makes Mash-up integration easier. However, there are some limitations of using the Mash-up technology, namely, limited Mash-up framework connector possibilities, scalability, stability and reliability of 3rd party API and reliability, and legal issues. The objective of this paper is to assess capabilities of Mash-ups in developing integrated applications and to identify their application areas. The application integration example using Mash-up possibilities is provided. It shows that using today's map Mash-up frameworks can be created web solutions that are more easily developed than using web service integration approach.

Keywords: mash-up, web-service, integration.

1 Introduction

The paper investigates integration of enterprise heterogeneous data sources using several integration approaches. The main two technologies are web services and Mash-up. Usually, the standard integration process using web-services and Business Process Execution Language (BPEL) is rather difficult and complex. Business processes usually are of dynamic nature. An enterprise wants to improve and modify, act in an agile manner, optimize, and adapt processes to improve the responsiveness of the whole company. Every change and improvement in a business process has to be reflected in applications that provide support for them. If we change one part of integration architecture, we must use reverse engineering in our integration approach. Modification of applications often is a difficult job, which requires time. Therefore applications cannot react instantly to changes in business processes—rather, they require some time to implement, test, and deploy the modifications. We must make information systems more flexible and adaptable to changes, and better aligned with business processes. There is the increased formality of composite applications, which are typically based on SOAP Web services and frequently woven together with BPEL and developed by professional programmers. Composite applications also tend to use an older generation of programming languages and technologies. Usually Web Service Composition heavily depends on the formalized standards and supporting environments, environments initially were comprehensible and required good programming languages knowledge. Mash-up is still a poorly defined and still misunderstood term. It is a term that is becoming popular to describe Web 2.0 sites that combine the features or functions of one website with another. In other words, there is a web 2.0 concept called "Mash-up", which allows consumers to draw upon content retrieved from external data sources (web services, data store and web application) to create entirely new and innovative services. [1,2] Mash-up essentially introduces a much simpler, more cost-effective, self-served approach for service composition, that significantly reduces the complexity and barriers of SOA service composition Therefore, every consumer can compose his/her own service applications only by "drag and drop" actions within a web browser. Obviously, Mash-up is a "consumer-centric" and lightweight service composition technology, which would be more applicable to all consumers all over internet in Web 2.0 paradigm. [3] If we look on Mash-up technologies from information technologies point, we can assume that Mash-up is based on Web principles and existing standards. Applying Mash-up technologies (likely, Web 2.0 related) in enterprise environment, we can produce low-cost, broad-sourced and rapid-growing applications to comply with the dynamic nature of business processes. Mash-up offer, technologies such as RSS (Really Simple Syndication), Web Services, and AJAX (Asynchronous JavaScript and XML), the Internet has facilitated the emergence of applications that are composed from a variety of services and data sources. Through tools such as Yahoo Pipes, these "Mash-ups" can be composed in a dynamic, just-in-time manner from components provided by multiple institutions. However, when using these applications, it is not apparent where data comes from or how it is processed.

The next section introduces specification of Mash-up technologies, types and usage and potentially best selection of Mash-up solution for an enterprise. The following sections will describe possibility of Mash-up usage in enterprise as a data presentation application and will give some example of data integration using Mash-up approach for enterprise. Some conclusions are pointed out at the conclusion section.

2 Types of Mash-ups

Mash-ups have several different colloquial interpretations what has resulted in some confusion regarding the term and its use. The technology industry extended this definition to encompass a new application genus that describes the combination of two or more sources into an integrated site. Basically, the Mash-up approach architecture consists of three components: Mash-up Engine, service providers and web browser (Figure 1).

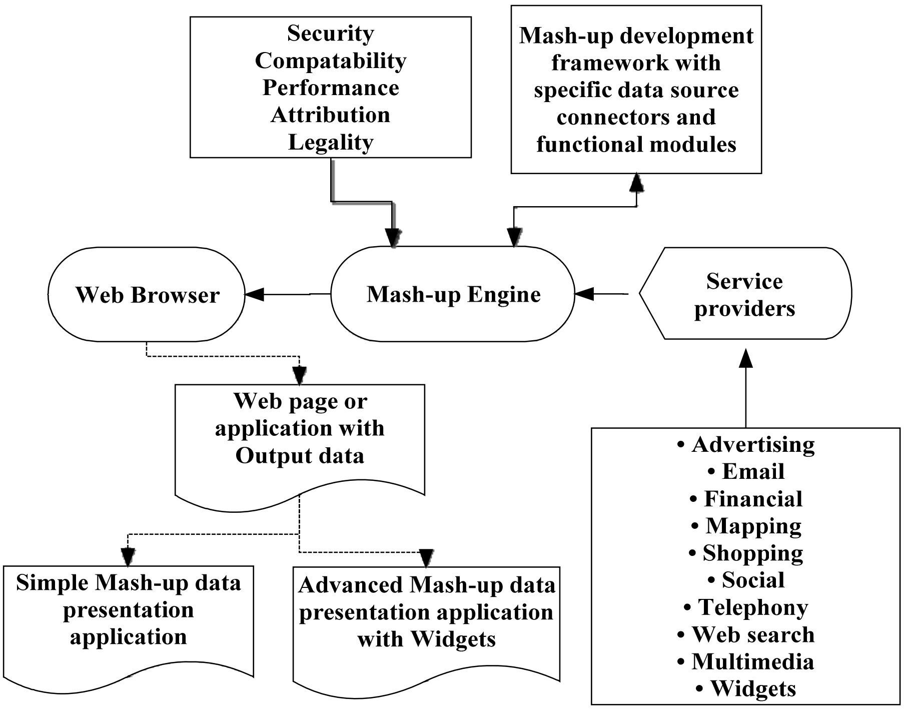

Security
Compatability
Performance
Attribution
Legality
Mash-up development framework with specific data source connectors and functional modules
Web Browser
Mash-up Engine
Service providers
Web page or application with Output data
• Advertising
• Email
• Financial
• Mapping
• Shopping
• Social
• Telephony
• Web search
• Multimedia
• Widgets
Simple Mash-up data presentation application
Advanced Mash-up data presentation application with Widgets

Figure 1. Architecture of the Mash-up approach

The typical types of Mash-up division we can see Table 1. The most common Mash-ups involve maps, but there are also video Mash-ups, photo Mash-ups, search and shopping Mash-ups, and news Mash-ups. Website developers can use data feeds and application programming interfaces (APIs) provided by established sites such as Google, Yahoo, Microsoft, Amazon, Ebay and others, which are created specifically to encourage Mash-ups. In other cases, Mash-up creators will use unauthorized tools such as screen craping to "borrow" content for their sites. In some cases, developers can use Widgets created from separate Mas-ups. We can assume that this solution as a three layer composition. Widget is created from composition of Mash-ups and finally this Widget is used to create a new Mash-up. Definitely it will be easier and easier to create new composites application, based on third party applications, therefore this approach can be used to create flexible data presentation applications.

Table 1. Types of Mash-up and specification

Type of Mash-up	Specification
Mapping Mash-ups	Combine data from different sources with a mapping program, such as Google Maps or Yahoo Maps. APIs for Yahoo Maps, MapQuest and Microsoft's Virtual Earth shortly followed, making it almost trivial to plug a rich source of geographical, topological, street-level and satellite image data into existing websites [5]
Video and Photo Mash-ups	Allow the combination of photos and/or video with the metadata stored with the image/video. Metadata can include information such as who took the picture, the date and time taken, etc. Mash-ups in this category may create a personal collage of photos by combining images available from a variety of sources and filtering those to only show ones that meet certain values of metadata – such as “where it was taken.” [4]
Search and Shopping Mash-ups	Are used to provide comparison shopping information. These types of Mash-ups typically combine data about product prices and availability. Additional data that might be integrated includes customer satisfaction ratings, shipping costs, and tax information.

Type of Mash-up	Specification
News Mash-ups	RSS feeds are used by many blogs and news organizations as a means of distributing (or syndicating) news headlines and story summaries. So that makes it possible for Mash-up developers to create personalized newspapers that meet your interests. Other Mash-ups combine keyword from news stories with maps and photo sharing sites. [4]
Consumer Mash-ups	Consumer Mash-ups are generally associated with Web 2.0. They require a lesser amount of programming expertise because they rely on public Web sites that expose well-defined APIs and feeds. The output is usually created by one of the sites participating in the Mash-up. [4]
Enterprise Mash-ups	Enterprise 2.0 Mash-ups (sometimes referred to as data Mash-ups) are more complex. Mash-ups are used solely by IT to rapidly deliver products. Application developers use both internal and external sources to create data Mash-ups and employ traditional coding techniques to create the user interface around them. This approach is the most difficult implementation to manage, but probably has the greatest impact. [4]

Architecturally, there are two styles of Mash-ups: web-based and server-based. Web-based Mash-ups typically use the user’s web browser to combine and reformat the data, server-based Mash-ups analyze and reformat the data on a remote server and transmit the data to the user’s browser in its final form. Also Mash-ups are defined in some generally accepted ways: presentation, process and data Mash-ups. J. Jeffrey Hanson, author of "Mash-up Patterns: Designs and Examples for the Modern Enterprise," describes the three main categories in Table 2.

Table 2. Mash-up categories and specifications

Mash-up categories	Specification
Presentation Layer Mash-ups	Is the least sophisticated type of Mash-up according to the Forrester Research taxonomy because these Mash-ups maintain a unified view of discrete content from disparate information sources. Examples of this type of Mash-up include the News Mash-ups described above and portals that welcome RSS feeds. Widgets, JavaScript, HTML snippets, DOM manipulation. [6]
Data Mash-ups	Mash-ups that are integrated at the data level, whether it's integrating files, databases, external Web service APIs. [6]
Process Mash-ups	Are the most sophisticated Mash-ups in the Forrester taxonomy. These Mash-ups allow users to combine data sources with business processes
Logic-based Mash-ups	This class of Mash-ups combines browsable logic with REST and SO AP services into a new logic component.

According to the Mash-up integration category and type, we can divide Mash-up creation into three complexity levels. If we must create Mash-up using some presentation layer, we can easily connect sources, logical object and get working our Mash-up in relatively short time. When Mash-up use files from many data sources that have been generated from other data sources or applications, Mash-up complexity are growing. We must use different data connectors, configure them, also connector possibilities are limited, configure data transformations and data flow. Integration business processes using Mash-up, can be very complex and time consuming, and in some cases using other technologies, as example web services, are effectively.

The main Mash-up key aspects, benefits, challenges, opportunities in development of integrated data presentation applications can be seen in Table 3 and 4.

Table 3. Mash-up aspects and benefits

Aspect and benefit	Specification
Effective leverage of Web parts and the Global SOA	Mash-ups are generally built out of the bits, pieces, and services of other Web applications that already exist, adding code only when it can't be sourced from internal or external suppliers or to provide integration "glue" between the parts.
Simple, lightweight software models and services.	Mash-ups are typically built using techniques like cutting and pasting snippets of Javascript, using feeds and XML to connect the various parts together, and even one-line Javascript inclusions that can pull in and integrate an powerful external component
A focus on self-service and do it yourself (DIY)	Mash-up development can be just as much for everyday Web users as it is for professional software developers. Applications that could never have been justified on a build-vs.-buy perspective (and took too long to acquire to help) are now possible

Table 4. Mas-up challenges

Mash-up Challenges	Specification
Deconflicting the two major Mash-up models.	On the consumer Web we have the natural, emergent Mash-up phenomenon as numerous Web parts suppliers and their consumers try different strategies out and sometimes hit upon the models that work best. On the vendor side, enterprise Mash-up tools try to add missing enterprise context like security, support for local SOAs, and so on, as well as a Mash-up development model, usually by choosing a particular Web component standard and providing a visual IDE that makes it easier for the non-HTML savvy to create Mash-ups [7].
Not enough Web services exist in our enterprises or on the Web.	There just aren't enough Web services available to supply the data and back-end functionality to Mash-ups. In the meantime we need to quickly service-enable our silos of Web-based information if we are to fully exploit it. This challenge can now be partially mitigated with companies like Kapow and Yahoo with Pipes, which are providing just such tools to make this possible [7].
Security and identity need to be sorted out.	The most useful Mash-ups will involve Web-based creations that are powered with personal and business information. Most Mash-ups don't require (or support) logins that allow it to collect information from your private repositories of information [7].
Too many widget formats.	Visual tool support is important to fully enable Mash-up development and realize the productivity potential and this support requires a consistent widget format. But the proliferation of widget models (both ad hoc and formal specs) makes visual tooling expensive and time-consuming to implement [7].

Mash-ups are surely accelerating the focus on do-it-yourself (DIY) and self-service applications. The Mash-up opportunities are described in Table 5.

Table 5. Mash-up opportunities

Mash-up Opportunities	Specification
Defining the essential ingredients of a successful Mash-up ecosystem.	What kind of Web services do you need? How about adapters to legacy systems and content? Should you encourage Mash-ups to be the foundation for other Mash-ups? How do we guarantee compatibility and interoperability? [7]
Addressing the tension between the two major styles of integration.	Mash-ups rely on live pulls of code from your supplier and are a much more extreme form of combining our systems together. Models of integration have security, testing, and version control issues written all over it [7].
Providing effective "enterprise context."	Enterprise Mash-up tools need to have solid stories around single sign-on (SSO), LDAP, JSR168 (portals/portlets), legacy integration, management, monitoring, RSS strategy, etc. Most enterprise tools are still falling short in these categories and will likely not get broad adoption until they address them [7].
Distribution and consumption.	Many of the ideas that the consumer Web has come up with for Web parts to be highly viral, easily distributable, and eminently consumable are also important strategies that we must think seriously about moving into our SOA initiatives. That's because our internal SOAs are smaller versions of the very same ecosystem that we have seen form on the Web. Getting services adopted and used in the enterprise has been entirely too hard up until now and even many companies out on the Web are falling short of in terms of applying the latest low-barrier, viral distribution techniques for success and uptake [7].
SEO, analytics, page views are all challenged by the Mash-up model.	Mash-ups turn single Web pages into entire applications and all three of these Web application models have been slow to address some of the more important models and monetization strategies that power business on the Web [7].

In general, there are two Mash-up models that are quite different from each other. First is the Consumer Web Model, were we have the natural, emergent Mash-up phenomenon as numerous web parts suppliers and their consumers try different strategies out and sometimes hit upon the models that work. The second being the vendor Model, enterprise Mash-up tools try to add missing enterprise context like security, support for local SOAs, a Mash-up development model, usually by choosing a particular Web component standard and providing a visual IDE that makes it easier for the non-HTML savvy to create Mash-ups. These commercial Mash-up application models try to figure out a priori what will work best, and in this they may be making the mistake of ignoring the vast laboratory of the Web that is already proving out highly effective models for Mash-up parts and integration strategies on a large scale. The consumer web model thrives well in an open source environment there by enjoying a large skill pool, their major focus being on ease of usage, with very little attention to aspects like security, deployment issues, manageability etc,. On the other hand the vendor model prefer a confined environment to safe guard its business interest, this leads to limited skill pool, also this model need to invest heavily in addressing major concerns like security, deployment, upgrades, managing etc. These differences between the two models tend to prevent the movement or migration from one model to the other [8].

As we can see, types, specification and categories of existing Mash-ups are rather different. This technology are expanding and developing, there is very hard to define strictly definitions of everyone aspect of Mash-up technology. Usually these technologies are mixed to develop Mash-up solutions, therefore there is no well-defined borderline between custom made Mash-up types. Also, when some type of Mash-up is realized, it could not be the same, as defined in some classification. For now, more frequently Mash-up is used in some easy and medium difficulty web solutions as example blogs, news portals, and web pages with maps. The researches, which presents new paradigms and frameworks showing how Mash-ups can be integrated for enterprise needs are unique and created for some specific problem. Our idea of enterprise Mash-up data integration environment is observed in the next chapter.

3 Mash-up usage in Enterprise

The main challenges of traditional enterprise data and application integration are cost, time and low responsiveness. The traditional application integration approach relies on having control over the applications and modifying them, adding, for example, a web-service interface that exposes the desired functionality and data. This approach results in expensive, long-running development projects involving highly skilled programmers, with high risk and high architectural impact. At the same time enterprises are pressed on budgets with fixed or reduced resources, defining APIs and communication paths is a time consuming process. Web services deliver a standard for programmatic interfaces to applications once web services have been created. But, there is still the problem of implementation. Modifying an existing application to expose its functionality and data in a programmatic interface is still a considerable task, and is only possible when there is full control over the application source code.

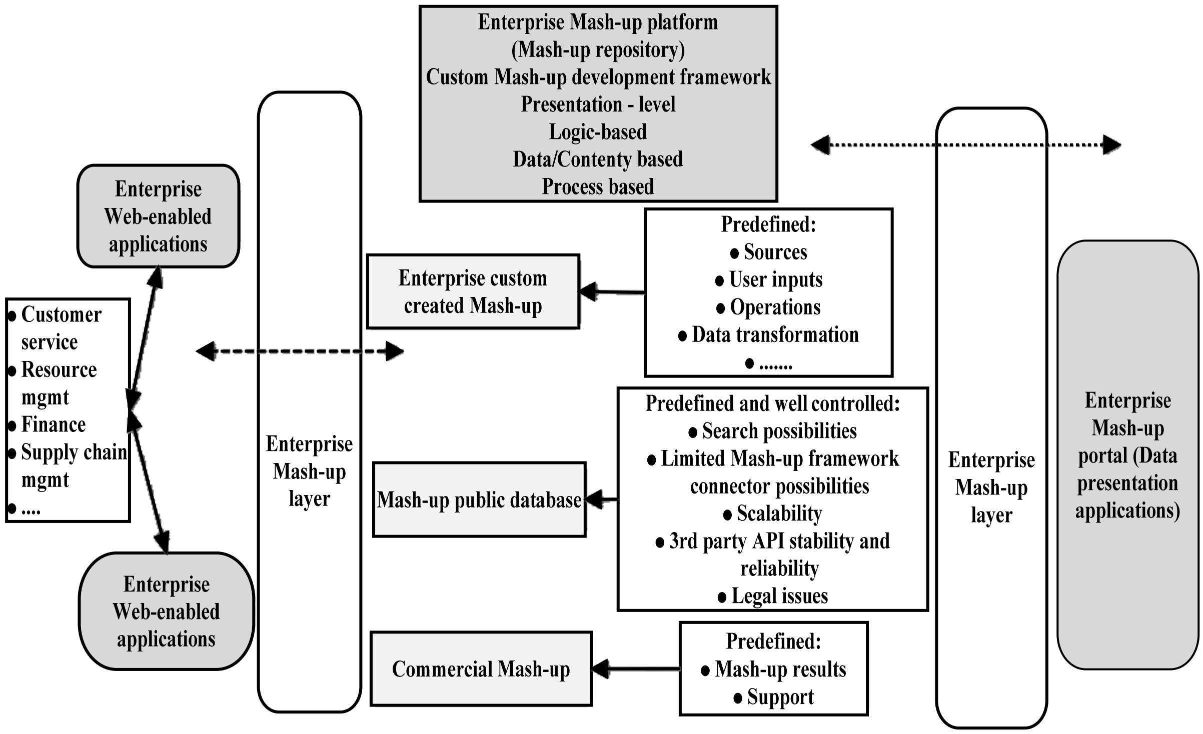


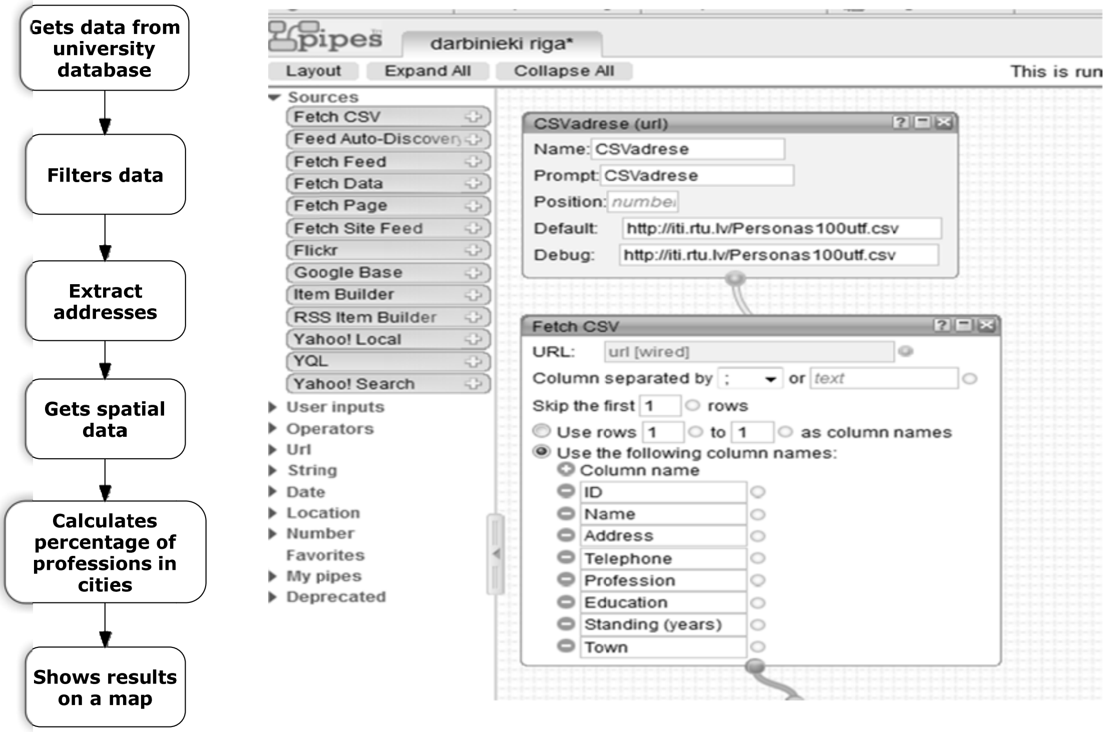
Figure 2. Enterprise Mash-up data integration environment

For applications outside of the company's realm, the modifications required by traditional integration methods will most often not even be possible. For the enterprise, the consequences include reduced productivity, increased operational costs, missed business opportunities and reduced business agility. One of the solutions is usage of enterprise Mash-up, that theoretically gives lower costs, non-intrusive integration, lower risk, shorter time-to-market, a faster and more accurate design phase, lower skill requirements (Figure 2).

Traditionally Mash-up is a web page or application that combines data or functionality from two or more external sources to create a new service. Enterprise Mash-up we can be defined as a layer that consists of data presentation application or applications to provide enterprise with specific data. Mash-ups are generally built out of the bits, pieces, and services of other Web applications that already exist, or data that already exists but is scattered around in multiple applications. If some of enterprise departments need some new information, using specific enterprise Mash-up layer, department can choose one of the Mash-ups or create new one and publish data into enterprise Mash-up portal. On the other hand possibilities of Business Process Execution Language and Mash-up development framework are very close. Mash-up development frameworks usually provide predefined data sources as example CSV, Google Base, RSS Item, fetch feed, data or page and etc. Also usually are available specific user input types. Data operators and string operators are also predefined by the framework environment. Comparison with BPEL, Mash-up provides basically all necessary things, but if solution is complicated, then usage of Mash-up can be more difficult and time-consuming than solution development using BPEL. Therefore Mash-up can be used to solve simplest need but for large-scale integration BPEL is better choice. By focusing on the simplest possible techniques and formats, enterprise Mash-ups approach appears to be successful and widespread primarily because just about anyone can create them across the enterprise. Mash-ups are typically built using techniques like cutting and pasting snippets of JavaScript, using RSS feeds and XML to connect the various parts together, and even one-line JavaScript inclusions that can pull in and integrate a powerful external component. While the source information is normally implicit on the web, a Mash-up allows relevant information to be integrated where existing web service architecture have potential problems, as example, scalability, performance, flexibility, implementability [9,10,11].

4 Application example

Application of the proposed enterprise Mash-up approach is demonstrated using an example of development of integrated data presentation application for finding qualified workforce using Yahoo pipes [12]. This example describes how easy is to create Mash-up map [5] needed for an enterprise to find qualified workforce in specific area. In fact, this application visually displays how many professionals live in each Latvian city. This system collects information from university database filters it and displays it on a map, as described in Figure 3.



CSVadrese (url)

Name: CSVadrese

Prompt: CSVadrese

Position: number

Default: http://iti.rtu.lv/Personas100utf.csv

Debug: http://iti.rtu.lv/Personas100utf.csv

Fetch CSV

URL: url [wired]

Column separated by ; or text

Skip the first 1 rows

○ Use rows 1 to 1 as column names

● Use the following column names:

Column name

ID

Name

Address

Telephone

Profession

Education

Standing (years)

Town

Figure 3. Integration steps

Prototypes functionality for proposed system is described as shown in Figure 3. The system uses CSV files [13] to get information from database. In this file graduates address, field of studies, and degree of studies are listed. “CSV input” module reads CSV file and interprets its containing information as an object. “Transformation” module renames objects, so further module could understand them. “Filter” module is part of a loop, it filters data based on professions and sends these records to “Counter” module which count how many people with given education are living in each city. “Chart creator” module uses Google char API to create pie chart of professions for each city. This module uses Google chart API by building an URL containing name of profession and numbers of each professions representatives. “Description creator” module uses the same information as “Chart creator” module, but it creates a sub-object that contains information about amount of profession representative in a city. As there are many people from on City it is necessary to remove duplicates, so there would be only one record for each City. This is achieved whit “Duplicate remover” module. “Location extraction” module is part of a loop. For each object this module looks at address field and using a geolocation service trying to find cities coordinates, if coordinates are found, this module adds sub-object “y:location” which contains coordinates. “Output” module displays the results. It can display results in tow modes as a list and on a map. For this module, to display results as a map, objects must have a sub-object “y:location”, which was added using “Location extraction” module. The final result is displayed as a map, where tags are added to every city and those tags contain charts whit information about profession distribution in that city (Figure 4).

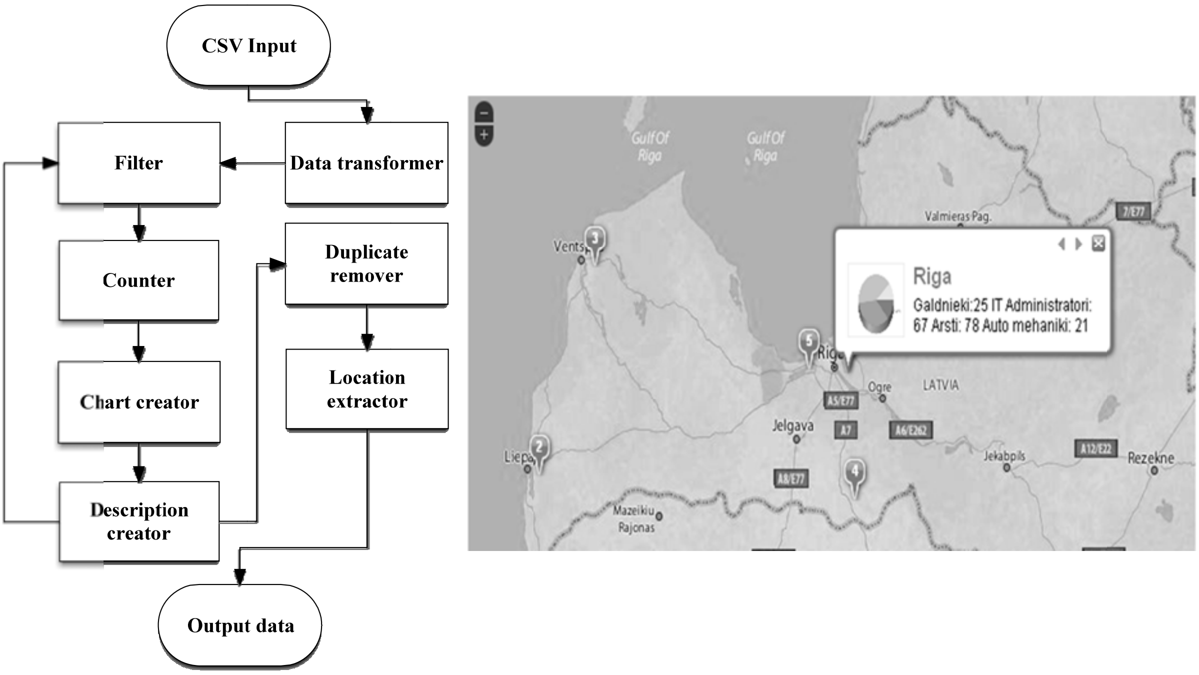


Figure 4. Prototypes working principle and Mash-up results

This system could be used by companies to visualize there costumer distribution so they could decide which products to advertise in which areas. A starting company could use this system to decide where to locate their business. A company could also use this system do decide how to distribute their employees over franchises. There are also possible applications in government’s different fields. State Employment Agency could use this system to help employers find qualified workforce in required areas. Also they could give suggestions for companies where to locate their businesses based on available workforce. State Revenue Service could use this system to visualize tax evaders and decide where to send their collectors. Now this system just shows how many people live in each city that can do certain jobs, when interface will be added company representative will be able to select a city and a profession. After selecting profession an interface will open where he can post a job opening, then this message will show up on employment agencies home page for that region.

This example of mapping, data and enterprise Mash-up combination is only one possible part of some enterprise portal, application, service or widget. Creating a similar data presentation application using the BPEL approach would be more difficulty than using Mash-up data connectors and functional modules, that are almost fully configured and user must only choose type, operations and finally connect them. Unfortunately, if solution will be more complicated with complex data selection, Mash-up standard functional modules do not provide all necessary operand and logical operations.

5 Conclusions

Enterprise portals will become more flexible and enterprise employees will be able to create custom Mash-ups to get necessary information more quickly than modifying business application functionality. Mash-up as an idea or technology is usable for development of integrated data presentation applications. Also, enterprise can use various standard portals as a presentation application. This is other technology, that include dynamic web parts, portlets, that can handle various tasks and display various information and in same time can be interactive and dynamically. These parts, portlets and widgets can actually be created using the Mash-up approach. The mash-up as approach has good future chances, but if only Mash-up will be used together with other enterprise information systems. Mash-up cannot fully replace BPEL and SOA ideology. Also, Mash-up applications can be evaluated according to their functional and non-functional requirements, for now, there is limited research done in this area. By increasing usage of Mash-up products, these requirements can be determinant for choosing best Mash-up solution.

For now, Mash-ups do not have well defined standards, data source connectors and operators. Also not yet available unique database, directories of Mash-up are also not available where users can find required Mash-ups, every Mash-ups service provider maintains own database.

References

[1] **Duane Merrill**. Mashups: The new breed of web app. IBM. 24.07.2009. [interactive 2010.03.12] access via the Internet: http://download.boulder.ibm.com/ibmdl/pub/software/dw/xml/x-mashups-pdf.pdf

[2] **Bob Rankin.** What is a Mashup? 2010. [interactive 2010.05.03] access via the Internet: http://askbobrankin.com/what_is_a_mashup.html

[3] **Xuanzhe Liu1, Yi Hui.** Towards Service Composition Based on Mashup. IEEE Congress on Services, Beijing, China. 2007.

[4] **Annette Easton, George Easton.** Demystifying Mashups. Proceedings of Informing Science & IT Education Conference (InSITE). 2010.

[5] **Guy Clapperton.** Mash-up maps. Engineering & Technology. 27 March - 23 April 2010. [interactive 2010.04.01] access via the Internet: www.theiet.org/magazine

[6] **Jeff Feinman.** The three types of Mash-ups. Software Development Times on the web. 07.16.2009. [interactive 2010.03.02] access via the Internet: http://www.sdtimes.com/blog/post/2009/07/16/The-Three-Types-of-Mashups.aspx

[7] **Dion Hinchcliffe.** Mashups: The next major new software development model? ZDNet. 14.06.2007. [interactive 2010.03.02] access via the Internet: http://www.zdnet.com/blog/hinchcliffe/mashups-the-next-major-new-software-development-model/106

[8] **Sunilkumar Peenikal.** Mashups and the Enterprise. Mphasis an HP company, white paper. 01.09.2009. [interactive 2010.04.07] access via the Internet: http://www.mphasis.com/pdfs/Mashups_and_the_Enterprise.pdf

[9] **Shah J Miah, John Gammack.** A Mashup Architecture for Web End-user Application Designs. Second IEEE International Conference on Digital Ecosystems and Technologies. 2008.

[10] **Dillon, T.S., Wu, C. and Chang, E.** "GRIDSpace: Semantic Grid Services on the Web:Evolution towards a SoftGrid", In proceedings of the Jrd International Conference on Semantics, Knowledge and Grid, Oct.2931,2007. Xi'an, China.

[11] **Kapow Technologies.** A Primer on Mashups in the Enterprise. 01.11.2006. [interactive 2010.01.23] access via the Internet: http://www.scribd.com/doc/6925245/A-Primer-on-Mashups-in-the-Enterprise

[12] **Yahoo pipes.** Documentation. Yahoo! Inc. 2010. [interactive 2010.04.10] access via the Internet: http://pipes.yahoo.com/pipes/docs

[13] **Kasper B. Graversen.** The CSV format specification. 2006-2009. [interactive 2010.07.20] access via the Internet: http://supercsv.sourceforge.net/csvSpecification.html