A Classification Method to Select a Mashup Creating Tool Based on Prior Knowledge of the End-User

Sofía Oraá Pérez

Grupo Reconocido de Investigación en Recuperación de Información y Bibliotecas Digitales (GRINBD)
Universidad de Valladolid
Email: sofiaoraa@gmail.com

María Mercedes Martínez-González

Departamento de Informática
Universidad de Valladolid
Edificio T.I.T., Campus 'Miguel Delibes' s/n, 47011 Valladolid
Email: mercedes@infor.uva.es

***Abstract*—Over the years, several tools and frameworks have appeared with the aim to facilitate the creation of mashups for the end user. These tools aim to integrate semantic and nonsemantic information available on the Web. However, not all users have the necessary technological knowledge to use them interchangeably. In this article, we propose a two stage classification for this set of tools that will allow users to select the most suitable tool based on their previous knowledge. In the first stage, we focus on the interface for the construction of the mashup provided by the system, enabling users to choose the technology they feel most comfortable with. In the second stage, a set of criteria for the choice of a particular tool are presented.**

Keywords–Mashups; Classification method; Selection of a tool.

I. INTRODUCTION

A mashup is a Web application that provides new functionality by combining, incorporating and transforming the resources and the services available on the net. These applications collect and process structured data from different sources and then display it for the users, while changing the original look-and-feel [1]. Mashups are particularly interesting because they facilitate the integration of information for a wide range of users. Thus, they become a way for users with little technical knowledge to perform this kind of tasks. An example of this technology is *Neighborhood Scout* [2] which allows users to select the best neighborhood to buy a house based on available data on schools, lifestyle or crime levels, among other factors.

As we have stated before, the purpose of this technology is to enable users to control relevant data, instead of software developers. However, the use of mashups involves programmers who must first study the data sources used to extract the necessary information in order to be able to reuse them. This task is quite complex, time consuming and it also undermines the main objective of this technology: to allow as much end users as possible to perform their own integration of information [1]. To overcome this problem, in recent years several tools and frameworks have been developed to facilitate obtaining information from different sources without the need to have developer knowledge [3].

Our objective is to help users without the necessary technological knowledge to choose the most suitable mashup creation tool. To do so, we provide a two-stage method of classification that facilitates the selection of a tool for mashups based on the user's prior knowledge. The existing classifications in the literature for these kinds of tools focus on a set of mashups, without a comprehensive review of them, or provide unclear and hard to apply criteria for the selection of a suitable tool when the user is inexperienced.

The first stage of our proposal classifies tools according to their interface, helping users to choose the tools with the most familiar technology from all available categories. Thus the number of tools available for the user will be limited, preventing them being overwhelmed and facilitating selection within a smaller set. The second stage provides a number of additional criteria that will help the user to select a specific tool from all available tools in that category. For example, a person who has studied statistics will be more familiar with using spreadsheets, so a tool using this technology will be the most appropriate one.

The rest of the paper is organized as follows: Section 2 contains a brief description of how this classification has been done in previous studies; in Section 3, we present the proposed method of classification; in Section 4 some of the existing tools are analyzed using the proposed method of classification; Section 5 contains an overall analysis of the tools studied; finally, Section 6 contains ideas for future work and the conclusions of our study.

II. STATE OF THE ART

1) Classification based on 4 criteria: Yu et al. [4] talks about the five most popular tools for building mashups in 2008. Two of them are no longer available. Whilst explaining their functionality, they enumerate their characteristics. This article divides the tools using four factors: the component model (type, interface or scalability), the composition (outputs, orchestration, data passing or ability to handle exceptions), the development (tool for inexperienced users or developers) and the runtime environment (browser plug-in or application stand-alone).

This article does not make a proper classification of the available tools. It simply makes a list of the characteristics of the five tools analyzed. Some of the criteria provided may be useful to some users; however, no further explanation is offered so as to understand or to extract concepts, meaning technical knowledge is required to use this classification.

2) Classification based on prior technical knowledge of the user: In 2009, Fischer et al. [5] divided the tools into six major types: programming paradigm, script language, spreadsheet, wiring, programming by demonstration and automatic creation. Their goal was to do a study of the tools to state if an inexperienced user could use them or if prior knowledge on

 ISBN: 978-1-61208-507-4

programming would be needed to do so.

Categories established in this classification do not serve as a prefilter based on user's profile to choose the right tool for two reasons: in each category they mix easy to use tools with more complex ones without proper clarification. In addition, the selected classification is not intuitive; that is, a user with no previous technological knowledge would not be able to use it for the selection of a concrete tool.

3) Classification based on the use of semantic knowledge: In his book *"A developer's guide to the semantic Web"* (2014) [1], Yu divided the tools into semantic and nonsemantic. The first ones to appear were the nonsemantic. An example of this technology are the "*map mashups*" [6] that allowed inexperienced users to exploit the usefulness of maps without having prior knowledge of programming or graphic mapping. These mashups had many limitations mainly due to the heterogeneity of the data; a change in the structure of this data forced them to reprogram their entire operation. That is why semantic mashups appeared, using Resource Description Framework (RDF) (as a data model) and SPARQL Protocol and RDF Query Language (SPARQL) (for task execution) allowing for effective organization, finding and representation of data regardless of the syntax. Therefore, they are better suited for change. An example of this technology is *Revyu* [7]. *Revyu* is a Web page to create reviews using RDF without having any knowledge of semantics.

Sorting mashups into semantic and nonsemantic does not provide enough information for people unfamiliar with this terminology; that is, for users without any knowledge of necessary technology, who will be the ones that will need more support to perform the selection of a suitable tool.

4) Classification based on 3 criteria: In her thesis (2014), Aghaee [8] developed a tool for creating mashups using natural language. She uses three criteria to classify existing tools: the usability of the systems based on the end-user's programming skills; how these users interact with the system and their various features; and the amount of aid provided by the system for the user to create their own mashup.

The classification focuses solely on mashups based on natural language, so it is insufficient to provide an overview of existing tools.

As has been observed during the evaluating of the existing classifications, none of them fits our objective and it is necessary to create a new classification method. The existing categorizations provide difficult to apply or unclear criteria, or else they are solely focused on a set of tools without an overall review. However, some of the supplied criteria can be useful and will be reused after being refined to achieve our goal: for a user to be able to select the right tool to develop mashups guided by our method of classification. To do this, we use a classification in two stages: first, using the interface provided by the tool for the user, which will determine the necessary knowledge about the technology that the user should have, and will allow users make a first filter selecting those tools that best fit their knowledge; the second, which will provide additional criteria allowing further refining of the selection by focusing on those features that make the tool unique.

III. Method of Classification. A proposal

A. First phase of classification

In the first stage of our classification, tools are divided based on the interface provided, allowing the user to select a set of them according to their previous knowledge. A summary of these categories can be found in Table I. What follows is a brief explanation of the selected criteria:

- Mashups tools **based on programming by demonstration** allow the users to generate their own mashup through a series of examples. It is the most appropriate technology for a person without previous knowledge of programming when they want to make the integration of information on pages whose structure hardly varies, for example, a news page. Their use is very simple. The user only needs to copy and paste pieces of the website indicating which content they want. After that, the tool will automatically be able to get the rest of the information on a particular topic following the structure indicated. Their functionality is limited.
- Mashups tools **based on databases** are very similar to those based on programming by demonstration, with the difference that they are able to adapt to changes in the structure of the Web pages consulted. This set of tools processes each of the existing Web pages as if it were a table in a database. It is therefore the appropriate technology for people without previous knowledge if the structure of the pages varies. These tools are able to integrate information following the HTML structure of the Web pages. As in the previous case, its functionality is limited, although it can adapt to the variation of the structure of the web making suggestions of possible integration. They require human intervention if the change in the structure is large.
- Mashups tools **based on widgets** contain graphic components used to create the mashup as an alternative to writing code. This type of technology is suitable for inexperienced users with little knowledge about technology who wish to do integration with more complex functionality than in the two previous cases. For example, the tool can be used to merge a single record in two different sources. Each widget is a black box with a specific functionality that can be used for integration of information even if the person does not know how the component works internally.
- In the mashup tools **based on pipes**, a "pipe" is used between each of the different data sources to connect them unifying their formats. As with tools based on widgets, these kinds of tools will allow the users with prior knowledge of programming to perform their own data integration using interchangeable components. This set of tools is more powerful, allowing the user to make adjustments, for example, changing the format during data integration or modifying the output structure. As a result, more knowledge is required to use these tools. They can be integrated into web graphics editors (based on widgets) to perform the most important tasks of integration and transformation of information [4].
- Mashups tools **based on spreadsheets** cannot access the real Web content, the information must be inserted directly into spreadsheets since it is the only input

TABLE I. CLASSIFICATION TOOLS: PHASE 1

Class	*Brief description based on the user profile*
Programming by demonstration	Inexperienced users. Pages with a stable structure.
Databases	Inexperienced users. Pages with varying structures.
Widgets	Inexperienced users with little knowledge about technology. Widgets with predefined functions that can be combined to achieve the integration of the data.
Pipes	Users with some prior knowledge of programming. Predefined functions that allow small adjustments using programming such as changes in the output format or in the structure of data integration.
Spreadsheets	Users with extensive knowledge in the use of Spreadsheets. Data input and output is in the spreadsheet where tasks are performed using predefined functions.
Scripting languages	Users with high programming skills. The user must develop the script that will perform data integration.
Automatic Creation	The system is able to obtain the data without human intervention.

format that these tools are able to understand. After processing the data, the results are inserted in the spreadsheet so the user can use them to draw their own conclusions. Their use requires someone with extensive knowledge on the use of spreadsheets, like someone who has studied statistics, for example.

- Mashups tools **based on scripting languages** are quite complex to develop, require a long time to create as well as high programming skills since it is the user who must create the script. That is why, inexperienced users cannot use this set of tools. Their use is recommended for programmers with extensive programming knowledge when they need to implement a very specific function that is not available in the easier-to-use tools.
- **Automatic creation** mashups tools include small components called *mashlets* to perform specific functions like automatically finding and proposing relationships between data without the intervention of human users. This type of tool is very useful when you have a well structured page with high semantic content as in the case of *DBPedia* [9]. Thanks to this technology numerous links between *DBpedia* and *RDF Book Mashup dataset* [10] have been automatically created [1].

After applying the first selection stage, we expect users to be able to find the type of tool that best suits their prior knowledge and with which they will be more comfortable. Table I summarizes the most appropriate set of tools depending on the user's profile. We expect that users will be able to use it to select the set of tools best adapted to their knowledge.

B. Second phase of classification

Within each of the categories listed in Table I we can find numerous tools, so in the second stage of our classification we propose a series of criteria to help the user to select a concrete tool among the ones available. A brief explanation of the selected criteria follows:

- **Autonomy of the tool**: this criterion is related to the ability of the tool to function as a complete program (stand-alone) that is installed on the end user's computer and will be able to function without an Internet connection; a Web tool that does not require installation; or a plug-in that must be installed on the user's browser.
- **Ease of use**: refers to the difficulty that the users will find to create their mashup. Although the users select the type of tool according to their prior knowledge, they may need additional programming skills to use a particular tool, thus preventing inexperienced users from using them. The tools can be easy to use, require prior knowledge of certain technology or require advanced programming skills.
- **Format of the data sources supported**: These tools can be designed to understand: the HTML structure of a Web page; The really easy to understand syndications (RSS) in XML used for sharing data on the Web; the information in various formats such as RDF, XML or CSV (Spreadsheets); or to read the set of subroutines, functions and procedures to facilitate obtaining the information grouped in the application programming interface Web (API).
- **Languages used**: there are many languages to express the content of Web pages that this set of tools can understand or use as output format to provide answers, including: RDF, SPARQL, XML, HTML, RSS, CSV or Atom.
- **License type**: the tools can be free (open source) or proprietary code, in which case it is necessary to purchase them.

TABLE II. CLASSIFICATION OF TOOLS FOR CREATING MASHUPS

Category	*Tools*
Programming by demonstration	ClipClip, Karma
Databases	Import.io, Yahoo Query Language, MashQL
Widgets	ClickScript, JackBe Presto Wires, Kapow, Lotus Mashups
Pipes	FeedsAPI, WebHookIt, Mulesoft, Huggin
Spreadsheets	Gneiss, StrikeIron SOA Express for Excel & Extensio Excel Extender for Microsoft Excel, AMICO:CALC, Open Refine
Scripting Languages	Web Mashup Scripting Language, WSO2 Application Server
Automatic creation	Revyu, Books@HPClab

The selected criteria can be useful as a guide to locate the right tool even in cases not covered by our work. We also hope that the criteria are clear enough to be understood even by those without experience in the field of computing.

IV. APPLYING THE CLASSIFICATION METHOD

Several examples of this technology exist, some of them can be found in Table II. To select these examples an exhaustive search with different keywords has been performed, including the terms: "mashups", "data integration tools" or "building mashups"; in several sources, such as: "Scopus", "Scholar" and "Web of Science". All selected tools are free or have a free trial version. We have focused on those tools with dates after 2012 that are available online for downloading and testing by users. As far as we know, this selection is broad enough to cover most available tools on the web.

Of all the tools available, one of each category described in the first phase has been selected, to which the classification method proposed will be applied. To perform the selection, the most recent date and the availability of an online tutorial to

 ISBN: 978-1-61208-507-4

TABLE III. CRITERIA FOR SELECTING THE MASHUP

Tool	*Category*	*Autonomy of the tool*	*Ease of use*	*Data sources format*	*Languages used*	*License Type*
Karma	Programming by demonstration	Stand-alone	Easy	HTML	HTML	Open Source
Import.io	Databases	Web Tool	Easy	HTML, JSON	CSV	Free
ClickScript	Widgets	Web Tool	Easy	RSS, HTML	JavaScript	Open Source
Huggin	Pipes	Stand-alone	Easy	API	RSS	Open Source
OpenRefine	Spreadsheets	Stand-alone	Easy	Spreadsheets CSV	RDF	Open Source
WMSL	Scripting languages	Web Tool	Advanced programming skills	HTML, metada, Javascript	WSLScripts	Open Source
Revyu	Automatic Creation	Web Tool	Easy	XML, HTML, RDF	HTML, RDF	Free

facilitate the installation and use has been taken into account. A summary of this information can be found in Table III.

Within mashups **based on programming by demonstration** we found: ***Karma*** [11] is a stand-alone, open source tool that allows the user to obtain, model and integrate data easily (Fig. 1) [12]. The user can see the result of the integration at any time during the creation process. This tool suggests predefined tags of its repository that could be helpful for the user to label their own data sets. For the extraction process it uses a Document Object Model (DOM) based on the structure defined by the user.

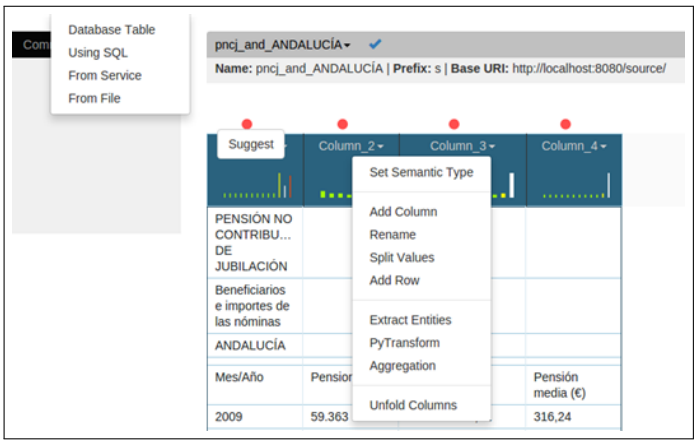

Figure 1. Karma.

As a tool for creating mashups **based on databases** we have: ***Import.io*** [13] is a free Web tool that allows users to obtain information from websites following their HTML structure [14]. It is very easy to use, the users just have to paste the URL that they want to study in the application. In addition, the tool provides a set of sample pages, videos, numerous documents and a forum where users can discuss the problems encountered. The results obtained after the extraction of information can be accessed and modified at any time. Similarly it is possible to add additional information to the data set. The user can choose the structure of integrated data and fill them with the appropriate information manually indicating the sections of the website to use (Fig. 2). Once processed, the data can be downloaded in CSV format.

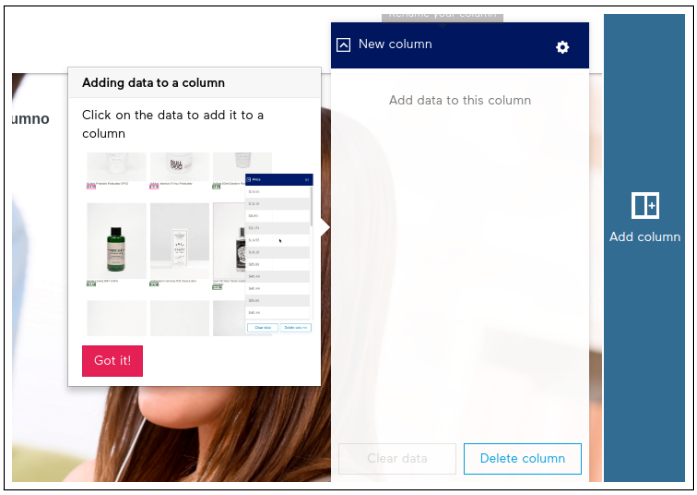

Figure 2. Import.io.

Among the tools **based on widgets** we want to highlight: ***ClickScript*** [15] is a free web tool to obtain data from RSS feeds and Web pages via Javascript functions [16]. The integration of data will be done through the Widgets available in the tool so that users without previous knowledge can use it easily. An example of these widgets can be seen in Fig. 3. The application provides information on the functionality of each widget, the necessary inputs and the outputs it provides.

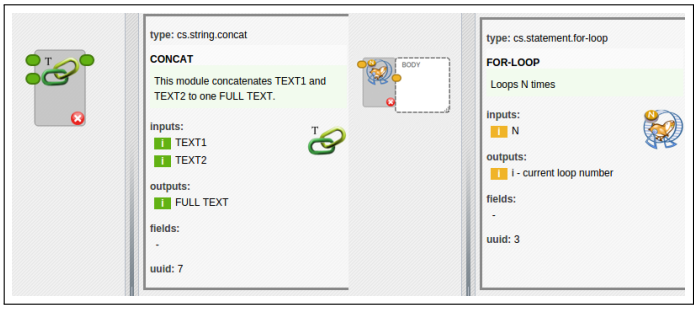

Figure 3. ClickScript.

The most popular example of **mashups based on pipes** was *Yahoo!Pipes*, a tool that was the basis for many other developments such as *DERI Pipes* or *Marmite*. Unfortunately, in 2015 the definitive closure of this tool was announced. As an alternative example we include: ***Huggin*** [17] is a free platform capable of connecting a lot of tools together. It uses the figure of the agent, i.e., a predefined functionality connectable with others. This tool takes advantage of the available APIs to connect applications such as Twitter, Dropbox, Basecamp or JIRA [17]. As an example of the functionality offered we have, among others, an agent capable of detecting changes in a document in Dropbox and sending them by mail; or an agent able to check the weather in a town and send an alert to the user's mobile at a specific time. It is easy to use,

 ISBN: 978-1-61208-507-4

the user must follow the instructions described on each agent (see Fig. 4). *Huggin* has a lot of online guides to help users during installation and use. Additionally it has the ability to add functionality by programming directly.

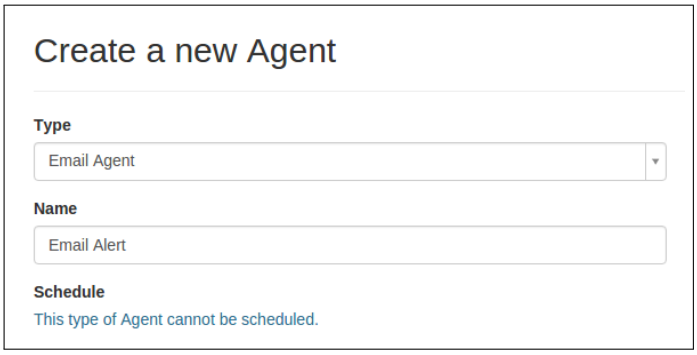

Figure 4. Huggin.

As an example among tools **based on spreadsheets** we found: ***Open Refine*** [18] is a stand-alone, open source tool developed by Google [19]. It adds functionality allowing the integration of information from different sources, such as, from one or more files from the user's computer, from a website via its URL or from a Google Drive document. It supports different formats, including CSV, Excel, XML and JSON. Open Refine also contains default features (Fig. 5) that facilitate management, integration and data filtering. This tool allows the user to add semantic information so data can be integrated in RDF format. The application is easy to use even for inexperienced users with minimal knowledge of spreadsheets.

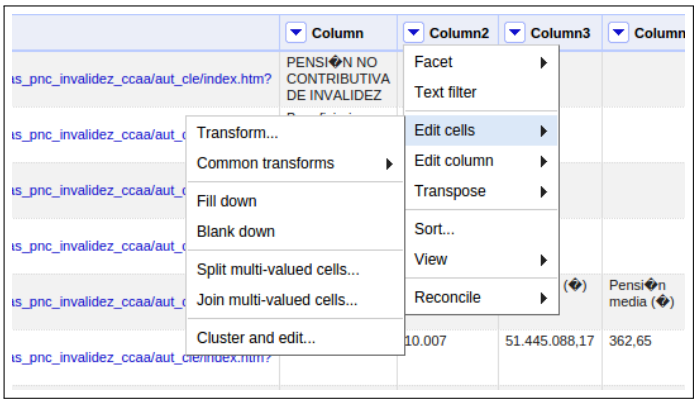

Figure 5. Open Refine.

An example of tools **based on scripting languages** is: ***Web Mashup Scripting Language*** o ***WMSL*** [20] allows end users to work on their browser without additional plug-ins [21]. To create the application the user must develop a page that combines HTML, metadata that describes the mapping relationships and a small piece of code or script, which is why advanced programming skills are required to use it. In the tutorial available online, numerous examples of this language appear (Fig. 6).

```
[extern] function function_name([argument1, argument2...])
{
  WMLScript statements here
  [return (some_value);]
}
```

Figure 6. WMSL.

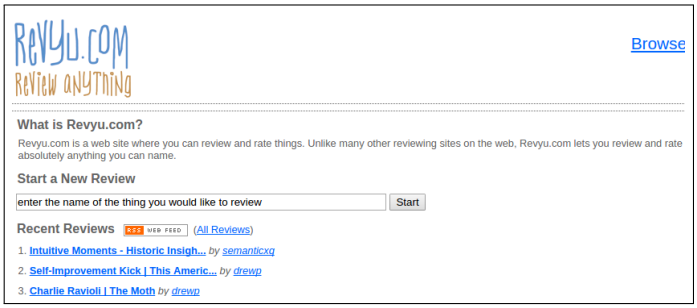

Figure 7. Revyu.

As an example of the tools **based on automatic creation** we have: ***Revyu*** [7] is a web application that allows users to create reusable reviews in RDF without knowledge of semantics being necessary [22]. This tool adds functionality allowing a user without the necessary knowledge to access, compare and query RDF sources. It is free and simple to use, the user must copy the URL of the website that he wants to analyze (Fig. 7). Revyu is able to understand HTML, XML and RDF. The integrated information will be presented to the user in HTML format so it can be easily read and in RDF format so it can be understood by a computer.

V. GLOBAL ANALYSIS OF THE PROPOSAL

Although the tools studied in the previous section can be used depending on the specific needs of users, the selection of the tool based on prior knowledge of the user is recommended. The classification proposal contains two stages:

The first stage classifies tools according to the interface provided. An inexperienced person interested in collecting information on a particular subject on a news website, should choose a mashup tool "based on programming by demonstration" that will allow them to obtain relevant data easily by simply copying and pasting the text from the page. The functionality of this tool is quite limited; however, it is the best technology to start using tools to create mashups. By contrast, people with extensive knowledge of programming who want information on an unknown and very specific topic will select a mashup tool "based on scripting language" that will allow them to develop their own functionality by programming their own script. This tool is quite complex to use and requires the user to know the structure of data to be processed perfectly. Even for expert users, it is recommended using simpler technologies like "based on widgets" for integrating information if the same functionality can be obtained.

The second stage of the classification method provides a set of criteria that will allow the user to select the specific tool within the category. One of the most useful criteria can be the "ease of use" when the user is inexperienced. However, it is possible that within a set of tools of the same category this criterion is not a differentiator. Before choosing the tool, the

user must take into account the "type of license". All the tools discussed in Section 4 of our article are free or provide a free trial version. Testing the tool before purcharsing it is recommended, to ensure it provides the proper functionality. Finally, the criterion on "the format of the data sources supported" indicates that the tool is able to understand different languages on the website without requiring the user to have additional knowledge and therefore without additional costs involved. If the tool is easy to use, has the necessary functionality for the integration of data and is able to understand the language in which the website is written, the user can choose any of the tools that meet the criteria.

Tools based on spreadsheets require input data to be inserted into the worksheet and the output will be generated in the same document using the same format. It is for this reason that the functionality may seem more limited when compared to other tools. However, experts in the use of this technology will find the ideal tool within this set.

In recent years more efforts have been made to develop tools that can understand natural language, i.e., the language used by humans to communicate. This language is quite complex to understand by a machine, that is why efforts to add semantics in Web services abound. With the increasing amount of linked data, ontologies and semantic information available on the web, it is likely that new "automatic creation" tools that do not require human intervention appear [23]. Unfortunately, in most cases, the heterogeneity of the sources prevents this automatic integration. Humans must evaluate the decisions taken. This field gradually progresses, however, progress is still not enough [24].

This classification is intended to help users, even those that do not have prior technological knowledge, to choose a tool even in cases not covered by our study. Generally, on the website of the tool users will be able to find the necessary information described in our classification criteria, which will help them to choose a proper tool.

VI. Conclusion and Future Work

This paper presents a classification method to help users choose a proper tool to create mashups. The fundamental criterion, complemented by others, that allows the refining of the choice, and that guides the proposal, is the prior technological knowledge of the user. The interest of a classification system that helps to select a tool is to reduce, the cost of this task for a user unaccustomed to building mashups, as it can become too costly and even frustrating in cases where the selection is wrong.

This classification is a step forward with regards to the ones presented in Section 2 because users, even those completely unaware of technological terminology, will be able to use it to select the right tool. This is clearly an advantage, since the proposed classification method may be used by a wider range of users. Of all classifications previously proposed, only [5] has a similar objective to the one laid out in this article, namely to help end-users choose a tool to create mashups. However, to understand and use our classification, minimal technical knowledge is required. Similarly, we hope that users can use this method of classification as a guide to locate the right tool even in cases not covered by our work: for example, tools that may arise in the future or existing ones that the users are able to locate on their own.

The application of our classification to select a tool is made in two stages, in order to simplify the work of selecting a tool restricting the number of tools to study. Thus, in the second stage the amount of tools will have been reduced to a reasonable number. The first phase classifies tools according to the interface they provide, allowing the users to select the technology they feel more comfortable with. The second phase of selection provides a set of criteria that will allow the user to select the most adequate tool.

Regarding future work, we propose to look into the use of at least a tool of each type of the ones discussed to perform the same task. An example would be: "The acquisition and integration of prices of existing degrees on the different university websites in Spain". To do this, the feasibility to perform the tasks will be studied using the different tools provided in this article.

Once we have ensured the viability to perform a certain task, an experimental study will be performed. Our classification method will be provided to different users in order to select the tool that best suits their prior knowledge. Thus, if after choosing the tool the user, regardless of his technical knowledge, is able to perform the task assigned, the usefulness of the classification method provided for selecting a tool to create mashups will be tested. The test of the classification method with real users will allow further conclusions to be drawn and new data to be generated.

References

[1] L. Yu, A developer's guide to the semantic Web, 2nd ed. Springer Science & Business Media, 2014.

[2] "Neighborhood Search for Home Buyers and Real Estate Investment," 2016, URL: https://www.neighborhoodscout.com/ [accessed: 2016-08-25].

[3] M. Krug, F. Wiedemann, and M. Gaedke, "Smartcomposition: extending web applications to multi-screen mashups," in Rapid Mashup Development Tools. Springer, 2016, pp. 50–62.

[4] J. Yu, B. Benatallah, F. Casati, and F. Daniel, "Understanding mashup development," Internet Computing, IEEE, vol. 12, no. 5, 2008, pp. 44–52.

[5] T. Fischer, F. Bakalov, and A. Nauerz, "An overview of current approaches to mashup generation." in Wissensmanagement. Citeseer, 2009, pp. 254–259.

[6] M. Batty, A. Hudson-Smith, R. Milton, and A. Crooks, "Map mashups, web 2.0 and the gis revolution," Annals of GIS, vol. 16, no. 1, 2010, pp. 1–13.

[7] "Revyu.com - Review anything," 2007, URL: http://revyu.com/ [accessed: 2016-08-25].

[8] S. Aghaee, "End-user development of mashups using live natural language programming," Ph.D. dissertation, Università della Svizzera italiana, 2014.

[9] "DBpedia," 2015, URL: http://wiki.dbpedia.org/ [accessed: 2016-08-25].

[10] "RDF Book Mashup," 2007, URL: http://wifo5-03.informatik.uni-mannheim.de/bizer/bookmashup/ [accessed: 2016-08-25].

[11] "Karma: Information Integration Tool," 2016, URL: https://github.com/usc-isi-i2/Web-Karma [accessed: 2016-08-25].

[12] R. Verborgh et al., "Survey of semantic description of rest apis," in rest: Advanced Research Topics and Practical Applications. Springer, 2014, pp. 69–89.

[13] "Import.io Lightning," 2016, URL: http://lightning.import.io/ [accessed: 2016-08-25].

[14] M. Butcher, "Import.io raises $13m series a for its data extraction platform," Tech Crunch, 2016.

[15] "ClickScript," 2016, URL: https://github.com/lnaef/ClickScript [accessed: 2016-08-25].

 ISBN: 978-1-61208-507-4

[16] R. Kleinfeld, S. Steglich, L. Radziwonowicz, and C. Doukas, “glue. things: a mashup platform for wiring the internet of things with the internet of services,” in Proceedings of the 5th International Workshop on Web of Things. ACM, 2014, pp. 16–21.

[17] “Huginn,” 2016, URL: https://github.com/cantino/huginn [accessed: 2016-08-25].

[18] “OpenRefine,” 2014, URL: http://openrefine.org/ [accessed: 2016-08-25].

[19] R. Verborgh and M. De Wilde, Using OpenRefine. Packt Publishing Ltd, 2013.

[20] “WMLScript,” 2016, URL: http://www.developershome.com/wap/wmlscript/ [accessed: 2016-08-25].

[21] B. Endres-Niggemeyer, The mashup ecosystem. Springer, 2013.

[22] A. Mayer, “Linked open data for artistic and cultural resources,” Art Documentation, vol. 34, no. 1, 2015, pp. 2–14.

[23] A. K. Kalou and D. A. Koutsomitropoulos, “Towards semantic mashups: Tools, methodologies, and state of the art,” International Journal of Information Retrieval Research (IJIRR), vol. 5, no. 2, 2015, pp. 1–25.

[24] R. Guha, D. Brickley, and S. Macbeth, “Schema. org: Evolution of structured data on the web,” Communications of the ACM, vol. 59, no. 2, 2016, pp. 44–51.

 ISBN: 978-1-61208-507-4