Linked Widgets: An Approach to Exploit Open Government Data

Conference Paper · December 2013

DOI: 10.1145/2539150.2539252

Ba Lam Do
TU Wien

Amin Andjomshoaa
Massachusetts Institute of Technology

A Min Tjoa
TU Wien

Linked Widgets - An Approach to Exploit Open Government Data

Tuan-Dat Trinh
Vienna University of Technology
Vienna, Austria
dat@ifs.tuwien.ac.at

Ba-Lam Do
Vienna University of Technology
Vienna, Austria
lam@ifs.tuwien.ac.at

Peter Wetz
Vienna University of Technology
Vienna, Austria
peter.wetz@tuwien.ac.at

Amin Anjomshoaa
Vienna University of Technology
Vienna, Austria
anjomshoaa@ifs.tuwien.ac.at

A Min Tjoa
Vienna University of Technology
Vienna, Austria
amin@ifs.tuwien.ac.at

ABSTRACT

More and more governments publish Open Government Data (OGD) for their citizens, which receives great interest; because they are not only covering many domains, but also are meaningful and trustful. Since the data are not ready to be linked to other sources, some governments have started to follow the *Linked Open Data* paradigm as a mean to improve data quality. However, exploiting Linked Data is still a challenging task, which requires a profound understanding of underlying datasets, as well as considerable technical expertise. In this paper, we present an approach to remove the barrier between end users and Linked Data generally, and Open Government Linked Data particularly.

Categories and Subject Descriptors

E.2 [**Data Storage Representations**]: Linked representations

General Terms

Design, Standardization

Keywords

Statistical data, Linked Open Data, Data Cube, Mashup, Widget

1 INTRODUCTION

The initial vision of the Semantic Web has been "*turning the World Wide Web to an environment in which information is given well-defined meaning, better enabling computers and people to work in cooperation*" [1]. However, only a small portion of information on the World Wide Web is presented in a machine-comprehensible way. Hence, in 2006, Tim Berners-Lee proposed the idea of Linked Data [2]. Linked Data as a set of best practices for publishing and connecting structured data on the Web is aiming to enable the evolution of the current Web of Documents into a Web of Data. In contrast to using hyperlinks in traditional HTML pages, Linked Data uses typed RDF links to represent relations, which helps datasets to be linked to each other and significantly increase the value of such datasets.

Attracting the scientific community, governments and policy makers, the research field of Linked Data has gained much momentum in recent years. We now have a huge amount of Linked Data that have been made available to the public. At the present, there are more than 300 datasets of cities, regions and governments[1]. However, exploiting Linked Data is still a challenging task, because it requires detailed knowledge of the datasets, as well as technical expertise. End users have no knowledge of the Semantic Web and cannot create SPARQL queries, which are used to extract information by themselves. As a result, applications that benefit from Linked Data are not yet as widespread as expected by its initiator Sir Tim Berners-Lee.

In this paper, we present an approach to remove barriers between end users and Linked Data, especially Open Government Linked Data. The underlying idea is the *Linked Widgets Platform*, which consists of many *simple* and *connectable applications*, so called *widgets*. Each application fulfills a simple task and can be linked to each other to enable new, more complex tasks. In other words, the output of one application can be the input of another one. Finally, without technical expertise or profound understanding of datasets, just by doing simple operations like drag & drop, end users can create new applications for themselves.

2 LINKED WIDGETS APPROACH

2.1 Introduction to Mashup

The idea of linked widgets is based on the "*Mashup*" concept. Mashup is defined as "*user-driven micro-integration* of Web-accessible data" [3]. The objective of mashups is to build effective and lightweight information-processing solutions based on accessible Web services of organizations. Mashups can be applied to a broad spectrum of use cases ranging from simple data widgets to more complex ones such as task automation and system integration [4]. The lightweight resource composition of mashups makes them appropriate for many domains such as business and enterprise applications, where different data sources can be combined together in a "quick and dirty" manner [5].

iiWAS2013, 2-4 December, 2013, Vienna, Austria.
Copyright 2013 ACM 978-1-4503-2113-6/13/12 …$15.00.

[1] http://datacatalogs.org/

Example of mashup applications are *Yahoo Pipes*[2] and *DERI's Semantic Web Pipes (SWP)* [6]. *DERI* is inspired by *Yahoo Pipes*, but it mainly supports semantic web data. It acts as a workflow that processes a set of RDF input data. The input data can be accepted in different formats, e.g., RDF, XML, Microformats, JSON and binary streams. The SWP framework supports users to query and retrieve data by sending the appropriate SPARQL queries to existing SPARQL endpoints, but users have to construct the queries by themselves. *DERI's SWP* solution provides an elaborate environment for composing Linked Data solutions based on well-defined operators; however, in order to work with the system, the end users should be familiar with Semantic Web technologies, and even Semantic Web experts have to spend a huge amount of time finding the appropriate datasets and formulating complicated SPARQL queries to collect their desired data.

2.2 Linked Widgets

Widgets are the conceptualization of a small processing unit for performing specific tasks such as fetching, parsing, formatting and visualizing data. They are implemented as full-fledged Web applications. In W3C's specifications, a widget is defined as "*an interactive single purpose application for displaying and/or updating local data or data on the Web, packaged in a way to allow a single download and installation on a user's machine or mobile device*" [7].

A *linked widget* is the evolution of widgets; each *linked widget* can solve specific problems. "*Linked*" means that widgets can be connected to each other so the output of a widget can be the input of another one. To end users, it is a black box, which has inputs and outputs. There is no need for special skills or advanced technical knowledge, since they do not have to understand how the widget retrieves the data or how the data is processed.

Linked widgets are our proposed key concept to exploit Linked Data effectively. If all users would be experts at Linked Data, we, hypothetically, would need only one SPARQL endpoint application. They could formulate desired SPARQL queries in order to retrieve data. But since this is not true, we need applications on top of the Linked Data layer, which satisfy each user's different information need. Since many users have many different information needs, many different applications fulfilling these needs are required. It is difficult to have enough resources to cover this requirement. But, as we propose, with linked widgets, the problem is at least partially solved. Suppose that we have five normal applications. So, we can solve only five specific problems. If they evolve into linked widgets, by making applications combinable, we can now create new solutions for new problems without programming. In most cases, the number of solved problems can be exponentially increased.

In *Yahoo* and *DERI pipes*, the concept similar to widgets is *operator*. *Operators* are fixed by their developers and are extended only by them, which restricts the applications to be more useful and powerful to end-users. Moreover, *Yahoo Pipes* is designed just for standard HTML/XML web data. *DERI Pipes* is better with semantic web operators (*SPARQL endpoint*, *RDF extraction*, *SPARQL result fetching*, etc.) but it is closed and it is impossible to be used by end users.

Linked widgets can be broken down into three main categories:

- **Data widgets**. These widgets are used as data feeds to other widget types and generate the required data in a specific format based on the given parameters. Examples of this type are data summaries, views, or SPARQL widgets.
- **View widgets**. The widgets belonging to this group generate visual output based on a given set of data at runtime [8]. Examples of such widgets include forms, diagrams, tables, maps and graphs [9].
- **Process widgets**. This group is dedicated to widgets, which receive a result from another widget as input and generate the output based on a customized process. Formatters, filters, and merge widgets are examples of process widgets.

2.3 Linked Widget Structure

There are three parts inside a linked widget. All of them are optional; it depends on the type of widget.

- **Input**: The input data of a widget.
- **Configuration**: The normal form fields or the options that affect the widget's processing. In other words, they can change the way the widget display data (if it is a *presentation widget*) or change the output of the widget (if it is a *process widge*t).
- **Output**: The output data that a widget returns after executing its function.

A widget can have multiple inputs, but **only one** output. To standardize the system, the input and output data of any widget is in form of JSON format.

2.4 Linked Widgets Platform (LWP)

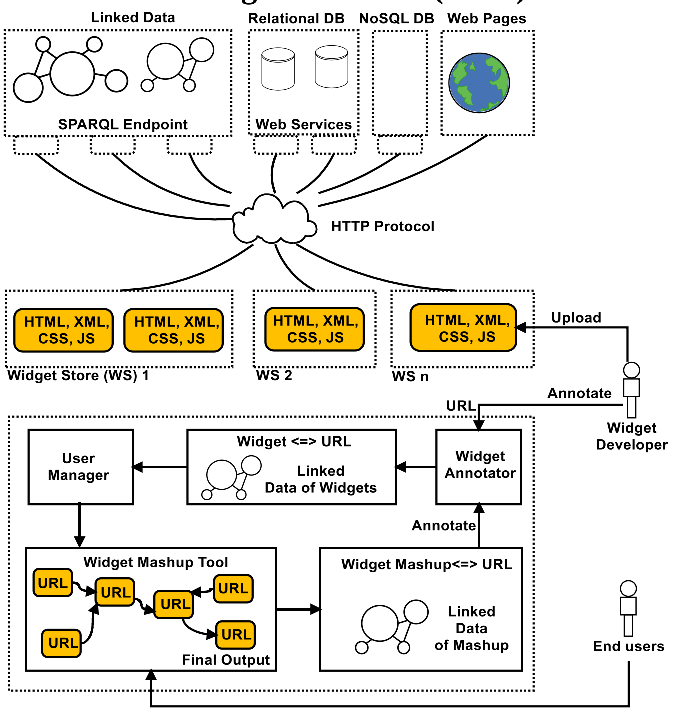


Figure 1: Architecture of Linked Widgets Platform

[2] http://pipes.yahoo.com/pipes/

Figure 1 shows the architecture of the LWP. There are three layers. The first is the *data layer*, which provides data to widgets through standard HTTP protocols. Though Linked Data is the primary data, a widget can take data from other sources, e.g., using JavaScript source code, it can access relational or NoSQL databases through web services. Additionally, a widget can directly parse an arbitrary HTML page to retrieve desired data.

The second layer is a series of physical widget stores. After developing a widget, a developer can upload it to any store, using any tool. Since widgets purely use HTML, XML, CSS and JS, it is very easy to deploy them. We encourage developers to develop the standard W3C widgets but it is not obligatory.

After uploading the widget to a store, developers have the URLs of their widgets. They now can officially join the LWP to share and broadcast their widgets. A tool called *Widget Annotator* in the third layer enables them to annotate their widgets in a structured and machine-readable way. As a result of this step, we have widgets based on a well-defined Linked Data structure. Furthermore, each widget has an URL as its reference.

Since showing all widgets to all users will be confusing, we propose a module called *User Manager*; Thanks to it, each end user has an individual list of widgets, which are available to do mashups in the primary tool called *Widget Mashup tool*. After creating a mashup, the user can retrieve desired data, without requiring any programming knowledge. All a user has to do is to understand the functions of available widgets, and then interconnect them by using drag & drop. After having connected several widgets the product is called a *Widget Mashup*. Each widget mashup is, again, identified by a URL. End users can enter this URL in the address bar of any browser to view the mashup.

To sum it up, the architecture here is dynamic and flexible. The system is open and can be extended as much as desired. It is a powerful tool, which enables all users to exploit Linked Data. Based on existing linked widgets and simple operations, they can create new widgets and new applications, which are easily shared by their URLs. The most complex part of this architecture is the "*Widget Mashup Tool*"; there all widgets reside and are run. By defining standard protocols, linked widgets are enabled to cooperate without knowing each other.

3 VIENNA OGD - A CASE STUDY

In Austria, Open Government Data was announced in April 2012 at data.gv.at. It provides many datasets of federal statistics, the states, cities, towns and useful information about transport, tourism, culture, and environment. We present a case study, which explains the conversion of this statistical data into Linked Data by following the *Data Cube Vocabulary*. Datasets from the following six categories containing statistical data in CSV format are converted: *Population*, *Management* & *Policy*, *Economy* and *Tourism*, *Transport* and *Technology*, *Working*, *Finance* and *Accounting*. This case study serves as an example to demonstrate how we use the W3C Data Cube Vocabulary to model Vienna OGD and convert it to Linked Data. Additionally, it is an example of how Linked Data can then be consumed in the proposed linked widgets system.

3.1 W3C Data Cube Vocabulary

The W3C Data Cube Vocabulary provides a means to publish multi-dimensional data, e.g., statistics data as Linked Data. The Data Cube Vocabulary is based on existing RDF vocabularies, e.g., SKOS for concept schema, SCOVO for core statistical structures, FOAF for agents, Dublin Core Terms for meta data, VOID for dataset description, and ORG for organizations [10]. If we consider the dataset we want to represent as a table in a database, each row in the table corresponds to an *Observation* in the vocabulary. The *DataStructureDefinition* class is used to describe the structure of the dataset**.** Using it, we can indicate which components are *dimensions* (similar to the key fields in a table), which components are *measures* (the remaining fields in a table) and which components are *attributes*. They are defined as follows: *Dimension components* serve to identify the observations. A set of values for all dimension components is enough to identify a single observation. *Measure components* represent the phenomenon being observed. *Attribute components* allow us to qualify and interpret the observed values. They enable specification of the units of measure, scaling factors and metadata.

3.2 Applying W3C Data Cube Vocabulary to Vienna OGD

3.2.1 Dimensions

Due to the variety of the data, each dataset can have different dimensions. For example, in the *Population category*, we have four dimensions: *Sex*, *Age*, *State*, and *Date* for the *b01 dataset*[3]. However for the *v01 dataset*[4], dimensions are: *District*, *Sex*, *Brand of car*, and *Year*.

3.2.2 Measures

In Vienna OGD, generally, we have more than 100 measures. Each dataset includes many columns. Some of them are dimensions and others are measures. There are different ways to decide between dimensions and measures. Consider an example of the *b22 dataset*[5] in the Population category shown in Table 1.

Table 1. An extract of the dataset b22

District	Pop_Men	Pop_Women	Date
90101	8050	8747	01.01.2012
90201	48146	49531	01.01.2012

Table 2. An extract of the dataset b22 after converting

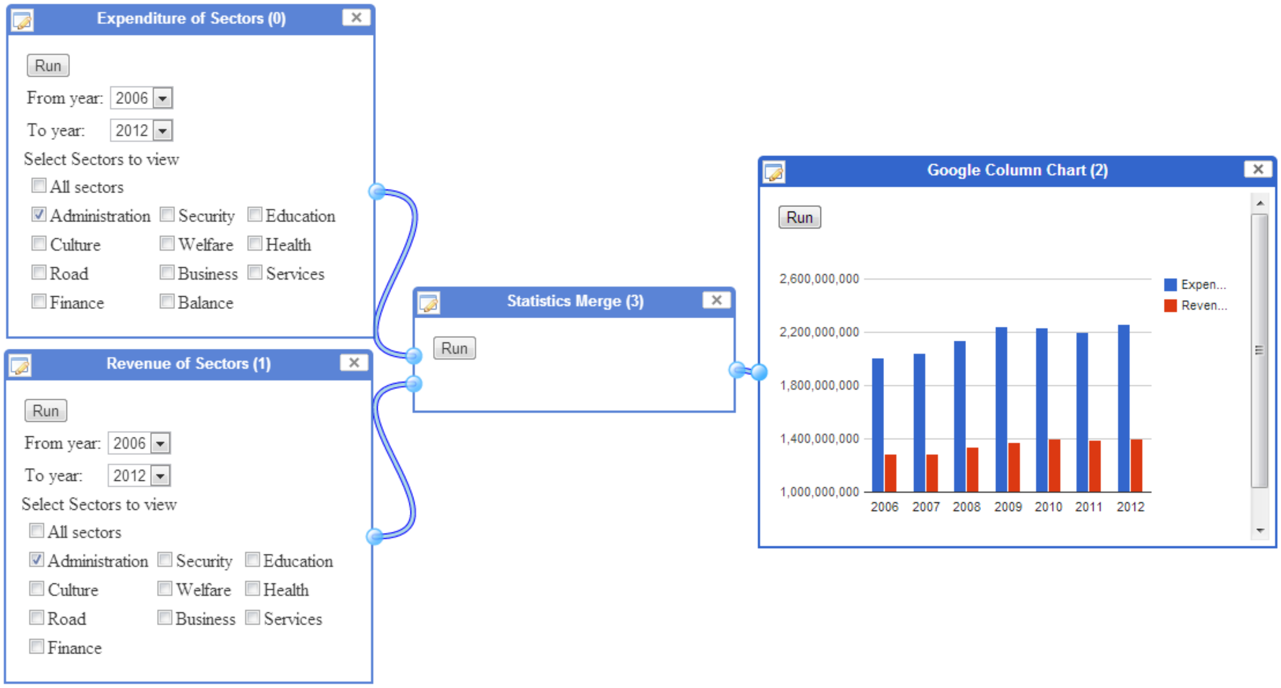
District	Sex	Pop_Sex	Date
90101	2	8747	01.01.2012
90201	2	49531	01.01.2012
90101	1	8050	01.01.2012
90201	1	48146	01.01.2012

Normally, we have two dimensions (*District* and *Date*) and two measures (*Pop_Men* and *Pop_Women*). However, to ensure that the dataset is connectable to other datasets, we create the *Sex dimension* which receives value *1* for *Men* or value *2* for *Women*. Two measures - *Pop_Men* and *Pop_Women* - are now converted into one measure, namely *Pop_Sex*, as illustrated in Table 2. The advantage of this processing is that the data is more meaningful, clearer, more coherent and therefore easier to link to other resources. However, as a consequence, the number of observations (equivalent to rows in a table) is doubled.

[3] http://data.wien.gv.at/katalog/bevoelkerung-geburtsbundesland-wien.html

[4] http://data.wien.gv.at/katalog/automarken-wien.html

[5] http://data.wien.gv.at/katalog/bevoelkerung-bezirke.html



Expenditure of Sectors (0)

From year: 2006

To year: 2012

Select Sectors to view

☐ All sectors
☑ Administration ☐ Security ☐ Education
☐ Culture ☐ Welfare ☐ Health
☐ Road ☐ Business ☐ Services
☐ Finance ☐ Balance

Revenue of Sectors (1)

From year: 2006

To year: 2012

Select Sectors to view

☐ All sectors
☑ Administration ☐ Security ☐ Education
☐ Culture ☐ Welfare ☐ Health
☐ Road ☐ Business ☐ Services
☐ Finance

Statistics Merge (3)

Google Column Chart (2)

Figure 2: A simple mashup visualizing administration expenditure and revenue of the City of Vienna

3.3 Results - Vienna OGD Linked Data

Following the goals of Linked Data, we focus on establishing both the links between datasets and links to other sources. In the first work, based on the conversion using Data Cube Vocabulary, datasets are connected by exploiting the similarity of dimensions and measures. For example, if two datasets have the same dimension(s), we can connect them together to compare common measures. Besides, we created other links between dimensions and measures. For example, there are links between the *sub-district* dimension and the *district* dimension. In the next step, we tried to manually link the data of districts, states, countries, etc. to external resources from *DBpedia* and *GeoNames*. As the final result, 140 links (*owl:sameAs)* are established.

From 70 CSV files, the resulting RDF file contains 1,817,503 triples. Next, for each dataset information about license[6], and for each observation information about its publisher (Vienna University of Technology) and source (Open Government Vienna) is added. As a consequence, the final RDF file contains 2,371,216 triples, which then is deployed on Virtuoso server[7] through a SPARQL endpoint[8].

3.4 Sample Use Case

The prototype version of the LWP[9] has already been developed. Compared to the architecture presented above, much more functions will be implemented in the future. However, at the present, end users can simply drag and drop widgets and inter-connect them in a web browser to retrieve desired results.

Figure 2 demonstrates a simple mashup based on four widgets. The goal of this mashup is to compare the expenditure and the revenue of any sector. Two data widgets are *Expenditure of Sectors* and *Revenue of Sectors*, collecting data by sending SPARQL queries to the Vienna OGD SPARQL endpoint. End users do not have to be equipped with semantic knowledge, since all semantic processes are hidden to them. The data from two widgets are merged via a process widget - the *Statistics merge*. Finally the results are visualized via a view widget - *Google column chart*.

Similarly, we can compare the expenditure and revenue between sectors by using two *Expenditure of Sectors* widgets or two *Revenue of Sectors* widgets. Besides, many more interesting use cases can be invented as desired by the user.

4 CONCLUSION

In this paper, we present the Linked Widgets approach to remove barriers between end-users and usage of Open Government Data. We also briefly present our work to model statistical data of Vienna OGD with the help of the W3C Data Cube Vocabulary and converting it to Linked Open Data as a case study.

Our main contribution is to introduce the novel concept of *linked widgets* and the platform that exploits them effectively. As a result, the solution providers and novice users will be able to create and adapt individual Linked Data applications based on interconnected widgets in a user-friendly environment without worrying about technical challenges of Linked Data consumption.

In future, we plan to implement the whole platform extended with additional useful functionality. Users will be able to only describe what they want and the system will suggest which widgets can be used or can automatically create a mashup for them.

[6] http://creativecommons.org/licenses/by/3.0/at/
[7] http://virtuoso.openlinksw.com/
[8] http://ogd.ifs.tuwien.ac.at/sparql
[9] http://ogd.ifs.tuwien.ac.at/mashup

5 REFERENCES

[1] T. Berners-Lee, J. Hendler and O. Lassila, The Semantic Web: Scientific American, Scientific American, may 2001.

[2] T. Berners-Lee. Linked Data - Design issues. http://www.w3.org/DesignIssues/LinkedData.html, 2013

[3] JackBe Corporation. 2008. A Business Guide to Enterprise Mashups. http://mdc.jackbe.com/downloads/JackBe_business_guide_to_enterprise_mashups.pdf, 2013

[4] G., Bader, W. He, A., Anjomshoaa, & A., Tjoa. 2012. Proposing a context-aware enterprise mashup readiness assessment framework. Information Technology and Management, 13(4), 377-387.

[5] A., Anjomshoaa, A. , Tjoa, & A. Hubmer, 2010. Combining and integrating advanced IT-concepts with semantic web technology mashups architecture case study. In Intelligent Information and Database Systems (pp. 13-22). Springer Berlin Heidelberg.

[6] D. Le-Phuoc, A. Polleres, M. Hauswirth, G. Tummarello and C. Morbidoni. 2009. Rapid prototyping of semantic mash-ups through semantic web pipes. In *Proceedings of the 18th international conference on World Wide Web*. WWW'09

[7] W3C. Requirement for Standardizing Widgets. http://www.w3.org/TR/widgets-reqs/ , 2013

[8] S. Wilson, F. Daniel, U. Jugel and S. Soi. 2012. Orchestrated user interface mashups using w3c widgets. In *Proceedings of the 11th international conference on Current Trends in Web Engineering*. ICWE'11

[9] A. Tahamtan, A. Anjomshoaa, E. Weippl and A. Tjoa. 2010. A SOM-based technique for a user-centric content extraction and classification of web 2.0 with a special consideration of security aspects. In *Proceedings of the 4th international conference on Knowledge science, engineering and management*. KSEM'10

[10] W3C. The RDF Data Cube Vocabulary. http://www.w3.org/TR/vocab-data-cube/, 2013