

## Leveraging the Web of Data via Linked Widgets

Amin Anjomshoaa, Elmar Kiesling, Dat Trinh Tuan, Do Ba Lam, Peter Wetz,

A Min Tjoa

*Institute of Software Technology and Interactive Systems, Vienna University of Technology,  
Favoritenstrasse 9-11, 1040 Vienna, Austria*

Received: 21 February 2014 / Accepted: XX Month 2014 / Published: XX Month 2014

© The Society of Service Science and Springer 2014

### ABSTRACT

Machine-readable datasets that have increasingly become available in open formats in recent years have great potential as a foundation for innovative applications and services. Linked Data in particular – a set of best practices for publishing and connecting structured data on the Web – has facilitated significant progress in evolving the Web of documents into a Web of Data. However, although this concept has opened up many opportunities for data sharing and collaboration, integrating data is still a challenging task that requires considerable technical expertise and a profound understanding of the underlying datasets. In this paper, we introduce a novel approach to provide knowledge workers with the necessary tools to leverage the fast growing Linked Data Cloud by creating semantic-aware dataflow processes. To this end, we introduce the “Linked Widget” concept as an enhancement of standard Web Widgets. These widgets are based on a semantic data model that facilitates powerful mechanisms for gathering, processing, integration, and visualization of data in a user-friendly Mashup environment. By allowing knowledge workers to easily create complex Linked Data applications in an ad-hoc manner, our approach should contribute towards reducing existing barriers of Linked Data adoption.

### KEYWORDS

Web of Data, Linked Data, Linked Widgets, Web Widgets, Mashups

---

Amin Anjomshoaa (✉), Elmar Kiesling, Dat Trinh Tuan, Do Ba Lam, Peter Wetz, A Min Tjoa  
Institute of Software Technology and Interactive Systems, Vienna University of Technology  
e-mail: { anjomshoaa, e.kiesling, dat, lam, peter.wetz, amin }@ifs.tuwien.ac.at

## 1. INTRODUCTION

The initial vision of the Semantic Web was 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 (Berners-Lee 2001). Even though Semantic Web technologies based on this idea have flourished, until recently only a small portion of information on the World Wide Web is presented in a machine-comprehensible way. Consequently, the lack of semantic information has hampered the realization of Semantic Web goals and initial expectations (Anjomshoaa et al. 2010). In response, Web specialists coined the term “Linked Data” (Berners-Lee 2006), which refers to a set of best practices for publishing and connecting structured data on the Web (Bizer et al. 2009). Following the Semantic Web approach, Linked Data uses the URIs to identify data sources on the Web and RDF (Brickley et al. 2000) to make typed statements that link those data resources to each other. In other words, Linked Data enables the Web of HTML documents and hyperlinks to evolve into a Web of structured and interlinked data. It provides a publishing paradigm in which not only documents, but also data can be a “first-class citizen” of the Web, thereby extending it to become a global data space based on open standards – the Web of Data (Heath 2011).

The Linked Data concept has gained considerable momentum in recent years. Today, the Web of Linked Data consists of vast datasets in various domains including media, geography, publications, government, life sciences, etc. Prominent examples include DBpedia<sup>1</sup>, Freebase<sup>2</sup>, PubMed<sup>3</sup>, and Linked GeoData<sup>4</sup>, which are all publicly available under an open license and shape the basis of Linked Open Data (LOD).

This development is supported by a growing number of Linked Data publishing tools, such as D2R Server<sup>5</sup>, RDF Refine<sup>6</sup>, and Pubby<sup>7</sup>, that handle technical details such as content

---

<sup>1</sup> <http://dbpedia.org/>

<sup>2</sup> <https://www.freebase.com/>

<sup>3</sup> <https://www.ncbi.nlm.nih.gov/pubmed/>

<sup>4</sup> <http://linkedgeodata.org/>

<sup>5</sup> <http://d2rq.org/d2r-server>

<sup>6</sup> <http://refine.deri.ie/>

negotiation and ensure that data is published according to the best practices of the Linked Data community (Bizer et al. 2009, Heath et al. 2008).

Despite the wealth of LOD resources and publishing tools, however, applications that utilize Linked Data are not yet as widespread as one might expect. An explanation for this situation is the difficulty of data gathering, processing, and integration, which imposes a significant burden upon end-users and knowledge workers within governments and enterprises. Efficient Linked Data consumption is hence still an open issue. Lowering the entrance barriers for data consumers could accelerate adoption of Linked Data among end users (Simperl et al. 2012; Bizer et al. 2012). This issue is particularly pressing because data integration is becoming vital in many areas of government and enterprise as organizations seek to benefit from open data and combine it with their internal information resources. In this context, we have identified a number of key issues that currently hamper effective consumption of Linked Data.

First, due to the open and distributed nature of LOD, datasets do not necessarily follow a uniform conceptual schema and frequently define concepts locally. As a result, a single real-world concept may have varying representations in different LOD datasets, which are typically not connected via appropriate schema-level links. This leads to datasets being only sparsely connected (Millard et al. 2010). Therefore, LOD developers need to familiarize themselves with details of each individual target LOD dataset and enrich it with cross-links in order to create data integration solutions, which is not a trivial process. In some cases, for instance, a single entity may be assigned different names in various vocabularies, which makes data integration a challenging task. As a result, the creation of user-generated applications requires considerable effort and results cannot be shared and reused efficiently.

Second, knowledge workers face a lack of custom-tailored LOD tools that support their specific requirements. This is similar to a general problem in many software domains such as business intelligence, where tools address around 80% of user requirements (Swayer 2008) and the remaining 20% are referred to as the “long tail of requirements”<sup>8</sup>. Current LOD tools

---

<sup>7</sup> <http://wifo5-03.informatik.uni-mannheim.de/pubby/>

<sup>8</sup> [http://en.wikipedia.org/wiki/Long\\_tail](http://en.wikipedia.org/wiki/Long_tail)

and frameworks typically address only generic requirements such as LOD publishing, storage, query, and reconciliation. They do not, however, cover the “long tail” of user requirements in the context of data integration and data processing problems. Therefore, the LOD community strives to develop multipurpose customizable tools to address the diverse requirements of novice end-users and developers.

Third, the lack of integration of LOD with Open APIs is another challenging issue. RESTful APIs (Fielding 2000) – one of the core Web 2.0 technologies – are based on a set of architectural principles by which Web services are structured around systems’ resources. Currently, many service providers incorporate Open APIs to expose their data and core functions for data integration, which facilitates lightweight service composition use cases. As is the case with Linked Data, a growing number of Open APIs (Weiss et al. 2010) today provide well-structured data in a scalable and resource-oriented manner. However, neither Linked Data nor Open APIs provide mechanisms for the integration of these two useful approaches in order to create more elaborate solutions (Verborgh et al. 2012). The LOD community has aimed to overcome this problem by introducing Linked APIs (Taheriyani et al. 2012). However, data integration environments still do not take full advantage of both LOD and Open APIs.

In this paper, we tackle Linked Data integration challenges by introducing a pluggable framework specifically designed to support Linked Data integration requirements. We introduce the concept of “Linked Widgets” as modular building blocks that augment the syntax and semantics of their underlying datasets and can be easily combined in a semantically aware Mashup environment.

The remainder of this paper is structured as follows: Section 2 provides background information and pointers to related work; Section 3 introduces Linked Widgets; Section 4 extends standard web widgets with a comprehensive semantic annotation and description model; Section 5 discusses the interaction of Linked Widgets within a Mashup environment, which leverages the semantic model to facilitate intra-widget communication. Section 6 provides details on a prototypical implementation and an illustrative example before we conclude in Section 7 with an outlook on future research.

## **2. BACKGROUND AND RELATED WORK**

Our approach mainly draws upon two streams of research: open data resources (including LOD and Open APIs) and mashup solutions. In this section, we explore these topics from a Linked Data perspective and outline the state-of-the-art.

### **2.1. Linked APIs**

In parallel with the growth of Linked Data resources, many other resources have been made exposed via dedicated services and APIs. This prevalent approach has had a strong impact on the evolution of data exchange and software architecture models. These services and APIs have driven the diffusion of the Web 2.0 and today, RESTful services – a scalable and resource-oriented architectural style – dominate the service Web. They have been accepted as the de-facto standard for distributed resource management in many modern Web applications and new services and APIs still emerge at an increasing rate.

Because both REST and Linked Data are based on core World Wide Web technologies, there are a number of common mechanisms – e.g., for naming, navigation, etc. – that can bridge the gap between resources that reside in these dispersed information spaces in a structured manner (Verborgh et al. 2012; Domingue et al. 2011). As a result, the Linked Data information space can be enriched with entities from the large body of resources behind services and APIs, which frequently deliver more up-to-date information. For instance, whereas the DBpedia information about a specific entity such as a company or person is updated periodically from Wikipedia, API services can deliver the updated results directly from the target data source. On the other hand, services and APIs can benefit from the connected nature of Linked Data resources.

A number of research efforts aim to bring together services and the Web of Data. Linked Services follow the principles of Semantic Web Services by annotating services and publishing those annotations as Linked Data (Pedrinaci et al. 2010; Pedrinaci et al. 2011). The Linked Services approach can benefit from existing reference service ontologies such as OWL-S, WSMO, WSMO-Lite and lightweight semantic annotation mechanisms such as SAWSDL, SA-REST, and MicroWSMO (Fensel et al. 2011a; Fensel et al. 2011b).

A complementary approach is to create services that consume and produce Linked Data. Linked Open Services (LOS), for instance, describe the input and output of target services as

SPARQL<sup>9</sup> graph patterns. Unlike traditional Semantic Web Services, *"the description of the service's input and output are no longer given by linking some value to some concept in the ontology, but rather by a (SPARQL) graph pattern which precisely describes the expected input graph and respectively the guaranteed knowledge content of the output statements"* (Krummenacher et al. 2010).

A third approach for combining services and the Web of Data is the Linked Data Services (LIDS) method, which provides HTTP URIs for entities (Speiser et al. 2010). Dereferencing the LIDS' URIs returns an RDF description of the service input entity, its relation to the service output and the output data itself.

## 2.2. Linked Data and Mashups

Mashups are defined as *"user-driven micro-integration of web accessible data"* (JackBe 2008). They owe their popularity primarily to Web 2.0 and SOA. The objective of mashups is to build effective and light-weight information-processing solutions based on Web services made available by various organizations. Mashups can be applied to a broad range of use cases that range from simple data widgets to complex task automation and system integration (Hoyer 2009; Koschmider et al. 2009). Lightweight resource composition makes mashups appropriate tools for many domains in business and enterprise contexts, where different data sources need to be combined in a "quick and dirty" manner. Major components of a mashup stack are APIs, widgets and mashups. In the context of Linked Data, there are a number of mashup solutions that are already part of the Linked Data cloud and make use of existing datasets. These mashups supply Linked Data with a consolidated set of RDF data that is gathered from distributed data resources and connected to the Linked Data cloud via appropriate links. The RDF Book Mashup (Bizer et al. 2007) is an example of such an application that demonstrates how Web 2.0 data sources such as Amazon, Google or Yahoo can be integrated into the Semantic Web. The RDF book mashup gathers information about books from different resources and presents the collected results as a consolidated resource with a specific URI in RDF format. Another example of a mashup application in the Linked

---

<sup>9</sup> <http://www.w3.org/TR/sparql11-query/>

Data world is the Bio2RDF tool (Belleau et al. 2008), which addresses the problem of knowledge integration in bioinformatics. Bio2RDF also combines data from multiple publicly available data resources from some of the most popular databases in bioinformatics. Bio2RDF joined the Linked Data cloud in 2009, and a number of successful applications have been created based on it. These mashup examples are designed for a specific purpose and work with a limited set of data resources, i.e., they cannot be customized and reused for other data integration scenarios.

Another group of mashup platforms aim to create dynamic solutions for consuming Linked Data. One example is DERI's Semantic Web Pipes (SWP) (Morbidoni et al. 2008, 2009; Le-Phuoc et al. 2009), a data integration tool for building data mashups based on existing Linked Data. The Semantic Web pipes act as a workflow that processes a set of RDF input data. The input data can be accepted in different formats such as RDF, XML, Microformats, JSON and binary streams. The SWP framework can also query and retrieve the required data from SPARQL endpoints. DERI's SWP solution provides an elaborate environment for composing Linked Data solutions based on well-defined widgets; however, in order to use the system effectively, end users must be familiar with Semantic Web technologies. Even Semantic Web expert users have to deal with complexities such as finding the appropriate datasets or formulating complicated SPARQL queries to extract the required data. In our proposed approach, such complexities are encapsulated in Linked Widgets and are hence abstracted from the view of novice end-users, who are instead presented with more high-level widgets. However, expert users can still gain access to more detailed components and implement their own customized widgets.

Paggr (Novack 2009) is another mashup-based approach that aims to simplify personalized aggregation of Linked Data resources via SPARQL-driven widgets. Unlike DERI's SWP, Paggr widgets are developed in a web-based environment to attract a larger number of developers. Paggr also uses the term "Linked Data Widget" to refer to its SPARQL-based widgets. Our Linked Widget concept, by contrast, is squarely centered on dataflow principles. Furthermore, Linked Widgets include a semantic graph-based model that distinguishes our approach from similar data integration solutions. This semantic model plays a key role for mashup creation and allows us to incorporate widgets as native resources into the LOD space.

### 3. LINKED WIDGETS

Although the plugin concept appeared as early as the mid 1970s<sup>10</sup>, the full power of such plugin-based platforms has only gained momentum with the advent of the Web 2.0 and the idea of crowdsourcing. Today, a large number of successful software platforms incorporate plugin functionality. Such platforms range from IDEs and Web browsers to mobile platforms and have proven to be very effective in addressing the “long tail” of user requirements.

One of the major goals of our approach is to introduce the plugin concept into the Linked Data domain. To this end, we introduce “Linked Widgets” as modular building blocks for the ad-hoc development of data-centric Linked Data applications. A Linked Widget is not only aware of the syntax and semantics of its underlying dataset, but also of the required metadata about provenance, statistics, and accessing methods. In other words, Linked Widgets are the conceptualization of a small processing unit for performing specific tasks such as fetching, parsing, formatting and visualizing data.

The key element of the specification of a Linked Widget is the definition of the required metadata that describes the widget, its Linked Data resources, and provenance information on its data. This metadata is made available to the public in the publishing phase.

Similar to web widgets, Linked Widgets are also implemented as full-fledged Web applications that are authored using Web standards. Such widgets either run completely on the client side or have a server-side counterpart in order to accomplish more complex tasks.

Our Linked Widgets are based on an extended and customized version of the W3C’s widget specification, which introduce the required additional information and functionality necessary to build Linked Data solutions. In the W3C 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*” (W3C Widget 2011).

The W3C Web Widgets specifications provide a generic framework that can be extended and reused for various scenarios (Chudnovsky et al. 2012). In line with these specifications and general Linked Data principles, a Linked Widget follows three principles:

---

<sup>10</sup> [http://en.wikipedia.org/wiki/Plug-in\\_\(computing\)](http://en.wikipedia.org/wiki/Plug-in_(computing))

1. Each widget and each mashup has a URI
2. User agents may dereference the widget via these URIs and obtain its semantic model that includes input and output data, provenance information, and the required underlying data resources.
3. A widget may have more than one semantic model, but all of them must generate the same output that must be explicitly linked to the input graph.

The varying semantic models represent input modalities that produce the same output. For instance, a *Film Widget* may accept various input modalities such as director name, actor name, or both. Then, depending on the selected variant, the appropriate query is formulated and sent to the corresponding data provider of *Film Widget* (e.g. DBpedia). In any case, the final output is always a list of movies.

Various types of Linked Widgets are necessary to support data acquisition, processing and visualization in data integration tasks. Accordingly, Widgets are categorized as follows:

1. *Data widgets* are used as data feeds to other widget types and generate data in a specific format based on a given set of parameters. Examples of this widget type include data summaries or SPARQL widgets.
2. *Process widgets* receive a dataset as input and generate the output based on a customized process. Formatters, filters, and simple merge widgets are examples of process widgets.
3. *Presentation widgets* generate visual output based on a given set of data at runtime (Wilson et al. 2012). Examples include diagrams, tables, and information visualization (Tahamtan et al. 2012).

Example Widgets for each category and their input and output data are presented in Figure 1. Data widgets play a key integrating role as interfaces to external data such as Linked Data and Open Data resources. Hence, the remainder of this section focuses on this widget type.

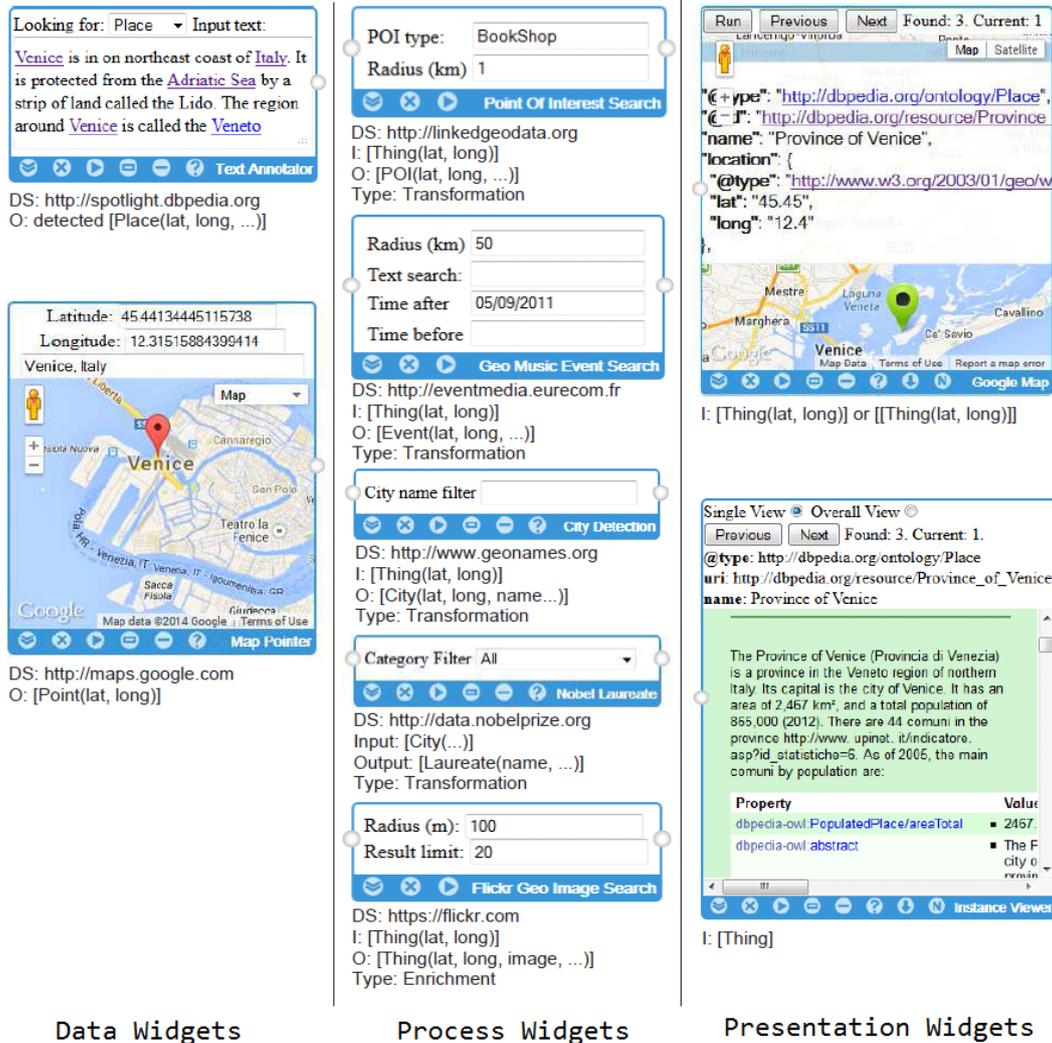


Fig. 1. Example Data, Process, and Presentation Widgets

Due to the open and distributed nature of Linked and Open Data, resources are published in heterogeneous formats and made available via a broad range of methods. For example, Linked Data can be downloaded and used locally or accessed via APIs and SPARQL endpoints. In order to spare end users the need to deal with such complexities, we wrap and abstract the steps necessary to access data resources in data widgets. SPARQL widgets, a subcategory of data widgets, for instance, send their embedded SPARQL queries to

predefined endpoints. Internally, the output of Data Widgets is always formatted as JSON-LD<sup>11</sup> which facilitates web-scale interoperability and interaction with other Linked Widgets.

Another use case for data widgets is the feeding of static data. A data widget may, for instance, provide sample data for big data resources, which may support and accelerate mashup development at design time. Furthermore, data widgets can deliver pre-calculated data, such as statistics, aggregations, etc., to add value to the target Linked Data resource. For example, a widget designer can use DBLP<sup>12</sup> data and create a data widget that provides the number of publications per journal/conference and per year statically.

#### 4. LINKED WIDGETS REPOSITORY

In order to promote sharing and reuse of Linked Widgets, they should be made an integral part of the Linked Data Cloud. To this end, they must be published on the Linked Data Cloud following Linked Data principles. In order for users to be able to explore, find and reuse them for their data integration mashups, a formal definition of information resources, metadata, and capabilities of Linked Widgets must be included in the published information.

To provide such a formal definition, we extend the semantic modeling approach of the Karma project (Taheriyani et al. 2012) for Web APIs and adapt it for the description of widget components. The resulting widget models connect inputs and outputs via a semantic model and can be used by widget providers to publish Linked Widget metadata.

Figure 2 depicts our Karma-based extended ontology for Widget and Mashup modeling. Karma, in turn, reuses the Semantic Web Rule Language (SWRL)<sup>13</sup> vocabulary. By extending the Karma ontology, we created a comprehensive ontology for Linked Widget specification that provides the concepts and properties necessary to model structural aspects such as inputs, outputs, provenance information, etc.

---

<sup>11</sup> <http://www.w3.org/TR/json-ld/>

<sup>12</sup> <http://www.informatik.uni-trier.de/~ley/db/>

<sup>13</sup> <http://www.w3.org/Submission/SWRL/>

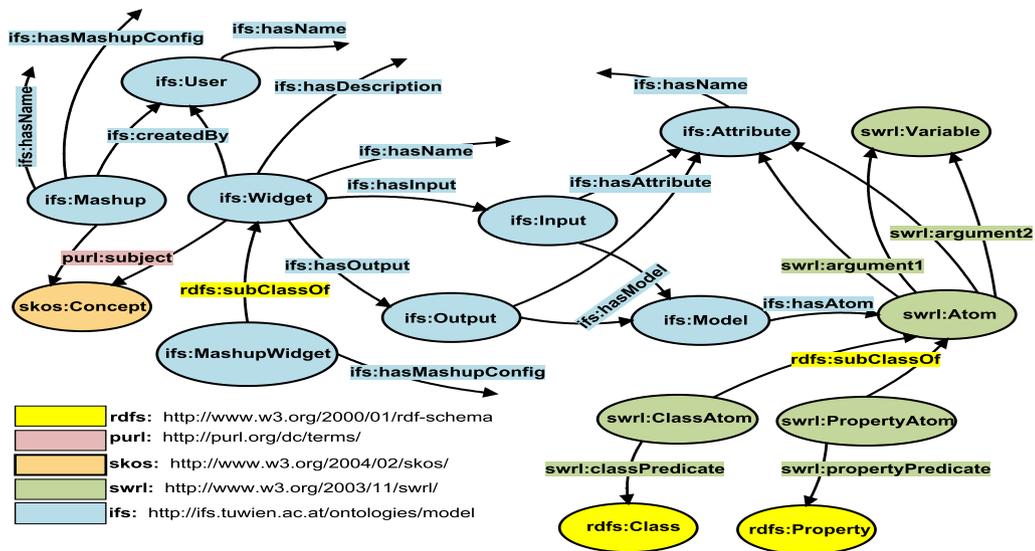


Fig. 2. Ontology for describing Widgets

Similar to Semantic Web services, mashups can benefit from the Semantic Web and ontologies to describe the profile, process model, and service grounding of their widgets. In the context of the proposed framework, a widget may query and discover other widgets that (i) produce data that can be used by this specific widget and (ii) are able to consume the data produced by this specific widget.

Using the semantic definition of a widget that includes ontology mappings for input and output parameters, the end user is notified of appropriate candidate input ports that can accept the results of a selected output port. In the context of Linked Data, semantic mashups can also leverage the underlying vocabulary (ontology) of a target dataset to provide added-value services for finding appropriate process widgets or even relevant data widgets that consume or provide the related data. The semantic description of widgets is stored in an RDF repository, which allows both end-users and assistive services to easily retrieve it via a dedicated SPARQL endpoint. Figure 3 depicts the human-readable serialization of a widget model for querying DBpedia for movies based on given criteria.

WidgetDBPediaFilmSearch at linkedwidgets.org  
<http://linkedwidgets.org/ontology/resource/WidgetDBPediaFilmSearch>

Property	Value
lw:hasAddress	./widgets/DBPediaFilmSearch/index.html
lw:hasDescription	Search and return LIST of Films from DBPedia
lw:hasName	DBPedia Film Search
is lw:hasWidget of	<ul style="list-style-type: none"> <li>lw:resource/DBPediaFilmWidgetCollection</li> <li>lw:resource/Mashupb5bd6c65a7e93d69f56d1c88aad2b83a_1395393958509</li> <li>lw:resource/Mashupb5bd6c65a7e93d69f56d1c88aad2b83a_1395394196056</li> </ul>
lw:hasWidgetModel	lw:resource/1399628005358_widgetModel1
lw:hasWidgetType	data
dcterms:subject	lw:resource/CategoryDBPedia
rdf:type	lw:Widget

**Metadata**

rdftype	<http://purl.org/net/provenance/ns#DataItem>
rdftype	<http://www.w3.org/2004/03/trix/rdfg-1/Graph>
foaf:primaryTopic	lw:resource/WidgetDBPediaFilmSearch
foaf:topic	Anon_0
<http://www.ontologydesignpatterns.org/CP/OWL/InformationRealization.OWL#realizes>	lw:data/resource/WidgetDBPediaFilmSearch
<http://purl.org/net/provenance/ns#createdBy>	Anon_1 (more)

[expand all](#)

This page shows information obtained from the SPARQL endpoint at <http://ogd.ifs.tuwien.ac.at/sparql>.  
[As Turtle](#) | [As RDF/XML](#) | [Browse in Disco](#) | [Browse in Tabulator](#) | [Browse in OpenLink Browser](#)

Fig. 3. A sample data widget model

## 5. LINKED DATA MASHUP FRAMEWORK

In this section, we introduce a mashup framework that acts as the hosting environment for creating and managing Linked Data-consuming applications. Figure 4 presents an overview of the proposed framework architecture and shows how the two major groups of end users – widgets developers and Linked Data consumers – can interact with the backend system.

Our framework benefits from recent advances in the mashup domain and provides a user-friendly environment for creating situational solutions. The repository of Linked Widget metadata described in the previous section is the foundation of this environment. The distributed nature of web widgets allows our Linked Widgets to be distributed on the web, i.e., there is no need to host all widgets that participate in a mashup on a single server.

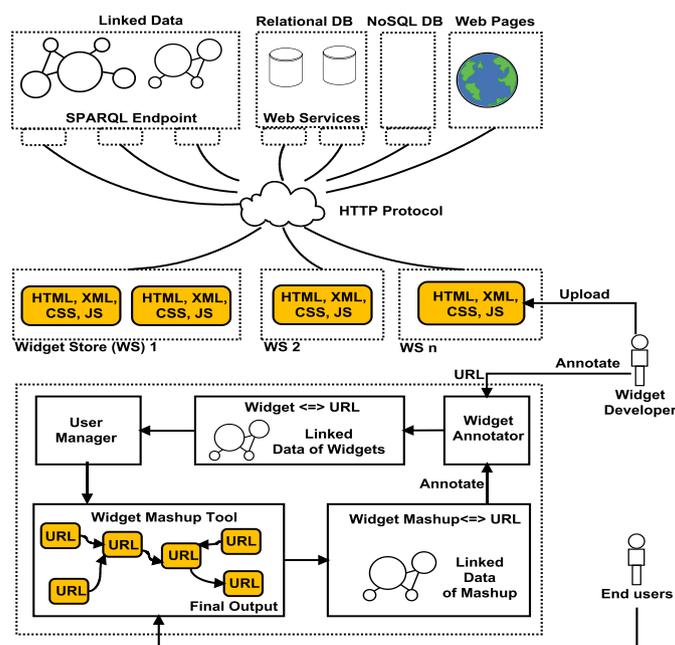


Fig. 4. Mashup framework for using Linked Widgets

A number of frequently used functions are provided as process widgets, which are included in the mashup framework and can readily be reused in mashup solutions. Basic visualizations, such as charts and diagrams based on services such as Google Charts<sup>14</sup> and Google Maps<sup>15</sup> are available as view widgets.

The mashup platform can integrate widgets from different repositories and integrate them in mashup compositions. All Linked Widgets have a unique URI that can be dereferenced to obtain meta-information. Furthermore, each mashup can be wrapped as a new widget with a unique URI. This wrapped widget is treated as and can be used like a regular widget.

## 6. PROTOTYPE IMPLEMENTATION

To demonstrate the feasibility of the proposed approach, we implemented the Linked Widget mashup framework and made it available to the public at the *linkedwidgets.org* website. The underlying data resources of this prototype include DBpedia, Flickr, and

<sup>14</sup> <https://developers.google.com/chart/>

<sup>15</sup> <https://www.google.at/maps/>

Geonames<sup>16</sup> as well as our dedicated SPARQL endpoint<sup>17</sup> that serves the Vienna's Open Government Data<sup>18</sup>.

The prototype implementation consists of three pillars. The first pillar is the implementation of Linked Widgets based on the specifications outlined. These widgets include data widgets such as *Points of Interest* (parks, swimming pools, etc.) and a number of simple data processing and visualization widgets such as *Geo-merge*, *Google maps*, *Google chart*, etc. Although, the widgets, and more generally mashup systems, are inherently reliant on the availability of the cloud services and underlying services (e.g., SPARQL endpoints), the distributed approach is attractive and redundancy mechanisms may improve the robustness of mashups in the future.

The second pillar realizes the mashup framework, which provides common mashup environment functionality as well as advanced Linked Data operations and functionalities. Finally, we used the ontology introduced in Section 4 for describing Linked Widgets and their corresponding user-generated mashups.

Figure 5 demonstrates a simple mashup created in the resulting mashup platform. It consists of four widgets and identifies locations available for returning a Citybike<sup>19</sup> (rental bike) in the city of Vienna. Furthermore, the mashup only returns locations that are less than 700 meter from a public swimming pool. To this end, we use two data widgets to retrieve swimming pools and Citybike stations with available return slots. Because these widgets are already available on our platform, novice end-users do not have to deal with the complexity of data retrieval from SPARQL endpoints, but can simply drag-and-drop the widgets into the mashup workspace and adjust parameters in the user interface. Next, the Citybike and public swimming pool locations are merged via a *Geo-merge Widget*, which returns points within the defined neighborhood. Finally, the results are visualized via a *Google Maps Widget*.

---

<sup>16</sup> <http://www.geonames.org/>

<sup>17</sup> <http://ogd.ifs.tuwien.ac.at/sparql>

<sup>18</sup> <http://data.wien.gv.at/>

<sup>19</sup> <http://www.citybikewien.at/>

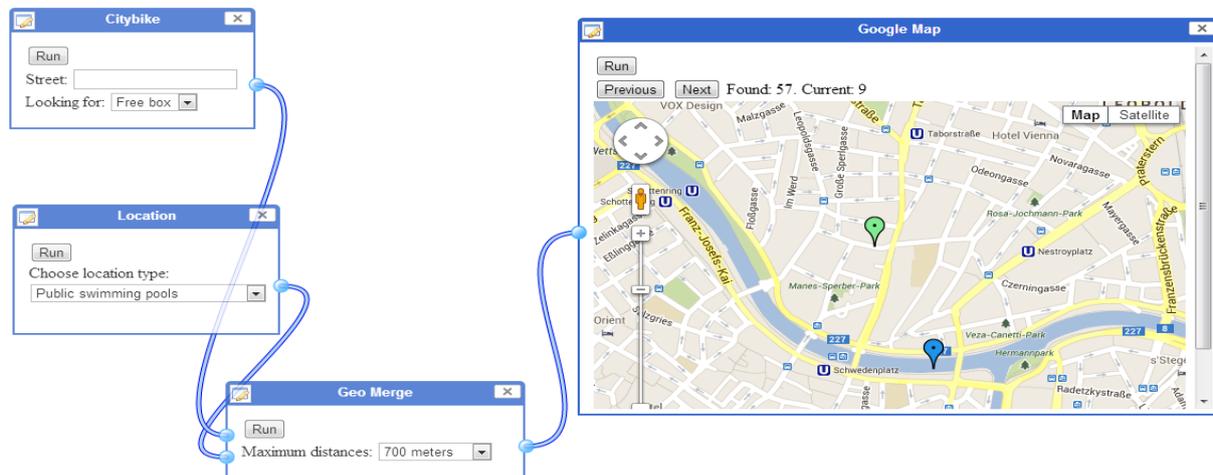


Fig. 5. Screen capture of a sample mashup based on Open Government Data of Vienna

## 7. CONCLUSIONS

In this paper, we tackled Linked Data integration challenges by uniting widget and mashup concepts in an integrated framework. In order to make Linked Data more accessible and useful to end users, we introduce “Linked Widgets” as building blocks of Linked Data mashup solutions. The key idea is to modularize functionality into widgets that can be shared, reused, and combined universally in various mashup solutions. This enables users to easily and dynamically obtain, enrich, transform, aggregate, and visualize data.

The proposed approach facilitates sharing and crowdsourcing of Linked Data widgets and mashup compositions. As a result, users, developers, and knowledge workers can create and adapt situational Linked Data applications in a user-friendly environment without having to deal with the technical intricacies involved in Linked Data consumption.

Our framework provides a formal method for publishing and sharing metadata on widgets and mashups following the Linked Data publishing principles. This metadata becomes part of the Linked Data Cloud, which allows users to explore, find and reuse widgets to address their varied data integration needs. The metadata is also used to implement some basic assistive services for widget-matching and widget auto-composition algorithms that are available in our framework. These algorithms make use of the annotated input and output models of the widgets, which is a unique feature compared to similar approaches.

Future research will focus on improving mashup compositions via elaborated assistive services based on the Linked Widget's semantic model. One such assistive service can be an advanced widget query service that is able to find alternative services in case some Linked Data resources are not available (e.g. due to network or maintenance problems) in order to enhance the systems availability and reliability.

## **ACKNOWLEDGEMENT**

Financial support by the Austrian Science Fund (FWF) under grant no P23122-N23 is gratefully acknowledged.

## **REFERENCES**

- Alexander K., Cyganiak R., Hausenblas M., Zhao J. (2011) Describing linked datasets with the void vocabulary," W3C Interest Group Note, W3C, 3 March 2011. [Online]. Available: <http://www.w3.org/TR/void/> . [Accessed May 2014].
- Anjomshoaa A., Tjoa A M., Hubmer A. (2010) Combining and integrating advanced IT-concepts with semantic web technology mashups architecture case study, *Intelligent Information and Database Systems*, pp. 13--22, 2010.
- Belleau F. o., Nolin M.-A., Tourigny N., Rigault P., Morissette J. (2008) Bio2RDF: towards a mashup to build bioinformatics knowledge systems, *Journal of biomedical informatics*, vol. 41, pp. 706--716, 2008.
- Berners-Lee T., Hendler J., Lassila O. (2001) *The Semantic Web: Scientific American*, Scientific American, May 2001.
- Berners-Lee T. (2006), *Linked data-design issues*, Available: <http://www.w3.org/DesignIssues/LinkedData.html> [Accessed May 2013].
- Bizer C., Cyganiak R., Gauß T. (2007) *The RDF Book Mashup: from Web APIs to a web of data*.
- Bizer C., Heath T., Berners-Lee T. (2009) *Linked data-the story so far*, *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 5, pp. 1--22, 2009.

- Bizer C., Boncz P., Brodie M. L., Erling O. (2012) The meaningful use of big data: four perspectives--four challenges, *ACM SIGMOD Record*, vol. 40, pp. 56--60.
- Chudnovsky O., Müller S., Gaedke M. (2012) Extending Web Standards-Based Widgets towards Inter-Widget Communication, *Current Trends in Web Engineering*, pp. 93--96.
- Domingue J., Pedrinaci C., Maleshkova M., Norton B. , Krummenacher R. (2011) Fostering a relationship between linked data and the internet of services, *The future internet*, pp. 351--364, 2011.
- Fensel D., Facca F. M., Simperl E., Toma I. (2011a) *Semantic web services*, Springer.
- Fensel D., Facca F. M., Simperl E., Toma I. (2011b) *Web Service Modeling Ontology, Semantic Web Services*, pp. 107--129, 2011.
- Fielding R. T. (2000) *Architectural styles and the design of network-based software architectures*, University of California, Irvine.
- Heath T., Hausenblas M., Bizer C., Cyganiak R., Hartig O. (2008) *How to publish linked data on the web*.
- Heath T., Bizer C. (2011) *Linked data: Evolving the web into a global data space, Synthesis Lectures on the Semantic Web: Theory and Technology*, vol. 1, pp. 1--136, 2011.
- Hoyer V., Stanoevska-Slabeva K. (2009) *Towards a reference model for grassroots enterprise mashup environments,* in *17th European Conference on Information Systems*.
- JackBe Corporation (2008) *A Business Guide to Enterprise Mashups*, JackBe, 2008.
- Koschmider A., Torres V., Pelechano V. (2009) *Elucidating the mashup hype: Definition, challenges, methodical guide and tools for mashups*.
- Krummenacher R., Norton B., Marte A. (2010) *Towards linked open services and processes, Future Internet-FIS 2010*, pp. 68--77.
- Le-Phuoc D., Polleres A., Hauswirth M., Tummarello G., Morbidoni C. (2009) *Rapid prototyping of semantic mash-ups through semantic web pipes*.
- Maali F., Erickson J. (2014) *Data Catalog Vocabulary (DCAT)*, [Online]. Available: <http://www.w3.org/TR/vocab-dcat/> . [Accessed May 2014].
- Millard I., Glaser H., Salvadores M., Shadbolt N. (2010) *Consuming Multiple Linked Data Sources: Challenges and Experiences*, in *Proceedings of the First International Workshop on Consuming Linked Data*.

- Morbidoni C., Polleres A., Tummarello G., Le Phuoc D. (2007) Semantic web pipes, Rapport technique, DERI, vol. 71, pp. 108--112.
- Morbidoni C., Le Phuoc D., Polleres A., Samwald M., Tummarello G. (2008) Previewing semantic web pipes, *The Semantic Web: Research and Applications*, pp. 843--848.
- Nowack, B. (2009). Paggr: Linked Data widgets and dashboards. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(4), 272-277.
- Pedinaci C., Domingue J. (2010) Toward the next wave of services: linked services for the web of data, *Journal of Universal Computer Science*, vol. 16, pp. 1694--1719, 2010.
- Pedinaci C., Kopecky J., Maleshkova M., Liu D., Li N., Domingue J. (2011) Unified Lightweight Semantic Descriptions of Web APIs and Web Services, 2011.
- Brickley, D., & Guha, R. V. (2000). Resource Description Framework (RDF) Schema Specification 1.0: W3C Candidate Recommendation 27 March 2000.
- Simperl E., Acosta M., Norton B. (2012) A semantically enabled architecture for crowdsourced Linked Data management, *CrowdSearch 2012*, pp. 9--14.
- Speiser S., Harth A. (2010) Towards linked data services," *The Semantic Web-Posters and Demonstrations, ISWC 2010*.
- Swoyer S. (2008) In Depth: The Non-Traditional Data Conundrum, in *Enterprise Systems Journal* [Online]. Available: <http://esj.com/Articles/2008/07/16/In-Depth-The-NonTraditional-Data-Conundrum.aspx?Page=3> , [Accessed October 2014].
- Tahamtan A., Anjomshoaa A., Weippl E. Tjoa A. (2010) A SOM-based technique for a user-centric content extraction and classification of web 2.0 with a special consideration of security aspects, *Knowledge Science, Engineering and Management*, pp. 412--423.
- Taheriyani M., Knoblock C. A., Szekely P. A., Ambite J. (2012) Rapidly Integrating Services into the Linked Data Cloud, in *International Semantic Web Conference*, Boston, MA, USA.
- Verborgh R., Steiner T., Van de Walle R., Gabarró Vallés J. (2012) Linked Data and Linked APIs: Similarities, Differences, and Challenges, in *Proceedings of the Ninth Extended Semantic Web Conference*.
- W3C Widgets (2011) Requirement For Standardizing Widgets, 27 September 2011. [Online]. Available: <http://www.w3.org/TR/widgets-reqs/> . [Accessed May 2014].

Weiss M., Gangadharan G. (2010) Modeling the mashup ecosystem: structure and growth, *R&D Management*, vol. 40, pp. 40--49.

Wilson S., Daniel F., Jugel U., Soi S. (2012) Orchestrated user interface mashups using w3c widgets, *Current Trends in Web Engineering*, pp. 49--61.

## AUTHOR BIOGRAPHIES



Amin Anjomshoaa is an Assistant Professor at the Institute of Software Technology and Interactive Systems (ISIS) of Vienna University of Technology and head of the Linked Data Research Lab. After receiving his Bachelor in Applied Mathematics, he has worked as Software Manager, Network Manager and Chief Technology Officer till 2003 when he joined the ISIS. He has received a master degree in Software Engineering and Internet Computing in 2005 and a second one in Information Management in 2007. In 2009 he obtained his PhD from Vienna University of technology and he is now pursuing his second PhD degree in Building Science. His research areas include Knowledge Management, Semantic Web, Linked Data, Web Engineering, and Building Information Modeling.



Elmar Kiesling is an Assistant Professor at the Institute of Software Technology and Interactive Systems (ISIS) of Vienna University of Technology, Austria. Furthermore, he is a senior researcher at Secure Business Austria, an industrial research center for IT security. He is a graduate of the school of Business, Economics, and Statistics at the University of Vienna, Austria, where he served as a researcher and lecturer and obtained a Master's degree in business administration and a PhD degree in management. His research interests include linked data, risk and security management, decision support systems, agent-based modeling and simulation, and visualization of multivariate data.



Tuan Dat Trinh is a PhD candidate at the Institute of Software Technology and Interactive Systems (ISIS), Vienna University of Technology, Austria. His research interests include Web standards and technologies, Semantic Web, and Linked Data. As a member of the Linked Data Lab at TU-Vienna, he is working on efficient data integration solutions for diversified and dynamic resources such as LOD and other open data sources.



Ba Lam Do is a PhD candidate at Institute of Software Technology and Interactive Systems (ISIS), Vienna University of Technology, Austria. He received his Bachelor and Master degrees in Computer Science from Hanoi University of Science and Technology (HUST), Vietnam. In 2008, he joined HUST as a teaching assistant and research assistant. In 2013, he received a grant from Vietnam-Austria Scholarship Programme for doing a PhD at TU-Vienna. As a member of the Linked Data Lab, he conducts his research in Linked Data with a particular focus on statistical data exploration from heterogeneous data sources.



Peter Wetz is a PhD candidate at the Institute of Software Technology and Interactive Systems of Vienna University of Technology. He obtained his Master's degree in Information Management at FH Joanneum University of Applied Sciences. In 2010 he gained first research experience in the field of Semantic Web at the Know-Center, a Competence Center for Knowledge-based Applications and Systems. At the Linked Data Lab of Vienna University of Technology, he currently he is focusing on Semantic Stream Reasoning techniques applied to the field of Environmental Informatics.



A Min Tjoa has been a full professor and director of the Institute of Software Technology and Interactive Systems at the Vienna University of Technology since 1994. He is the chairman of the Austrian National Competence Center for Security Research, vice-chairman of the IFIP Technical Committee for Information Systems, chairman of the IFIP Working Group on Enterprise Information Systems, and vice-chairman of the DEXA Association (Database and Expert System Applications). He has served as chairman of several international conferences including the IEEE Int. Conf. on Distributed Computing Systems (ICDCS), European Software Engineering Conference (ESEC), ACM SIGSOFT Symposium on the Foundations of Software Engineering

(FSE), and the International Conference on Electronic Commerce and Web Technologies (EC-Web). His current research focus areas are data warehousing, cloud computing, semantic web, security, and non-standard IT applications. He has published more than 200 peer reviewed articles in journals and conferences.