Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 813875, 12 pages
http://dx.doi.org/10.1155/2014/813875

Research Article

Semantic Information Integration with Linked Data Mashups Approaches

Hanh Huu Hoang,[1,2] Tai Nguyen-Phuoc Cung,[3] Duy Khanh Truong,[1] Dosam Hwang,[2] and Jason J. Jung[2]

[1] *Hue University, 3 Le Loi Street, Hue City 530000, Vietnam*
[2] *Yeungnam University, 280 Daehak-Ro, Gyeongsan, Gyeongbuk 712-749, Republic of Korea*
[3] *Hue Industrial College, 70 Nguyen Hue Street, Hue City 530000, Vietnam*

Correspondence should be addressed to Jason J. Jung; j2jung@gmail.com

Received 19 March 2014; Accepted 24 March 2014; Published 28 April 2014

Academic Editor: Chong-Gun Kim

The introduction of semantic web and Linked Data helps facilitate sharing of data on the Internet more easily. Subsequently, the resource description framework (RDF) is the standard in publishing structured data resources on the Internet and is used in interconnecting with other data resources. To remedy the data integration issues of the traditional web mashups, the semantic web technology uses the Linked Data based on RDF data model as the unified data model for combining, aggregating, and transforming data from heterogeneous data resources to build Linked Data mashups. There have been tremendous amounts of efforts of semantic web community to enable Linked Data mashups but there is still lack of a systematic survey on concepts, technologies, applications, and challenges. Therefore, in this paper, we investigate in detail semantic mashups research and application approaches in the information integration. This paper also presents a Linked Data mashup application as an illustration of the proposed approaches.

1. Introduction

The development of generic web applications is well understood and supported by many traditional computer science domains, such as classical database applications. In current web application development data integration and access are typically dealt with by fairly sophisticated abstractions and tools which support rapid application development and the generation of reusable and maintainable software components. The task of programming such applications has become the task of combining existing components from well-established component libraries, that is, customizing and extending them for application-specific tasks. Typically, such applications are built relying on a set of standard architectural styles which shall lower the number of bugs and ensure code that is easy to understand and maintain.

The emergence of semantic web and its extensions into the web has been making an excellent revolution of smart web applications and helps in information integration with different approaches. One of main trends in semantic web community is about publishing big datasets to the web in the format of Linked Data or raw RDF. This is about representing the data in form of interconnected resources. This is reflected of W3C's vision of the semantic web: (1) the semantic web is about common formats for integration and combination of data from diverse sources, where on the original web they mainly concentrated on the interchange of documents; (2) the semantic web is also about languages for representing how the data relates to real world resources. That allows a person, or a machine, to start off in one database and then move through an unending set of databases which are connected not by wires but by being about the same thing [1].

Historically, the process of writing new queries and creating new graphic interfaces has been something that has been left to the experts. A small team of experts with limited skillsets would create applications, and all users would have to use what was available, even if it did not quite fit their needs [2]. A *mashup* is an (web) application that offers new functionality by combining, aggregating, and transforming resources and services available on the web [2]. Therefore, mashups are an attempt to move control over data closer to the user and closer to the point of use. Although mashups are technically similar to the data integration techniques that preceded them, they are philosophically quite different. While data integration has historically been about allowing the expert owners of data to connect their data together in well-planned, well-structured ways, mashups are about allowing arbitrary parties to create applications by repurposing a number of existing data sources, without the creators of that data having to be involved [3]. Therefore, mashup enabling technologies not only reduce the effort of building a new application by reusing available data sources and systems but also allow the developers to create novel applications beyond imagination of the data creators. However, the traditional web mashups still suffer the heterogeneity of data coming from different sources having different formats and data schema. To remedy the data integration issues of the traditional web mashups, the semantic web technology uses the Linked Data based on RDF data model as the unified data model for combining, aggregating, and transforming data from heterogeneous data resources to build Linked Data mashups. Powered by tools and technologies having been developed by the semantic web community, there are various applications domains building applications with Linked Data mashups [4].

There has not been any work that gives a comprehensive survey about technologies and applications of Linked Data mashups as well as the challenges for building Linked Data mashups. This shortcoming comes from several following reasons. Typical Linked Data mashups are data-intensive and require the combination and integration of RDF data from distributed data sources. In contrast to that, data-intensive applications using RDF are currently mostly custom-built with limited support for reuse and standard functionalities are frequently reimplemented from scratch. While the use of powerful tools such as SPARQL processors takes the edge off of some of the problems, a lot of classical software development problems remain. Also such applications are not yet built according to agreed architectural styles which are mainly a problem of use rather than existence of such styles. This problem is well addressed in classical web applications. For example, before the introduction of the standard 3-tier model for database-oriented web applications and its support by application development frameworks, the situation was a lot similar to the situation that we see now with RDF-based applications [1].

This paper investigates research and application approaches in information integration using semantic mashups on Linked Data datasets and presents an application for demonstration purpose. This paper is structured as follows: following the introduction in Section 1, Section 2 is about background on semantic mashups. Sections 3 and 4 present an overview on semantic mashups architecture and approaches. Followed by Section 5, Linked Data mashup issues are also described in an analytical view. Open challenges are withdrawn in Section 6. Finally, a demonstration example using semantic pipes, Metaweb, and raw datasets for mashups purpose is presented. This paper is concluded by remarks and outlook for future work.

2. Background

2.1. Semantic Mashups. The emergence of semantic web and Linked Data helps in sharing structured data on the web easily. RDF has been recommended by W3C for a standard of publishing data on the web and linking different data sources. This trend has created a wave of raw data publications on the web and made it become open data sources with a highly semantic representation where individuals and organisation can share their data as an open format with each other. Thanks to this innovative trend, semantic mashup applications have been evolved and developed with these open semantic data sources.

Hence, semantic mashups is a mashups application using RDF as its background data model and SPAQRL for tasks execution [4]. By applying semantic web technologies into semantic mashups, we can organise, seek, and represent data in an effective manner to users.

2.2. Linked Data. The term *Linked Data* refers to a set of best practices for publishing and linking structured data on the web. These best practices were introduced by Tim Berners-Lee in his web architecture note, namely, Linked Data [1] and have become known as the Linked Data principles. These principles are descripted as follows: the basic idea of Linked Data is to apply the general architecture of the World Wide Web to the task of sharing structured data on global scale [4].

Linked Data principles firstly advocate using URI references to identify not only web documents and digital content but also real world objects and abstract concepts. These may include tangible things such as people, places, and cars or those that are more abstract, such as the relationship type of "knowing someone." Linked Data use the HTTP protocol for web resources access mechanism with the use of HTTP URIs to identify objects and abstract concepts, enabling these URIs to be "dereferenced" (i.e., looked up) over the HTTP protocol into a description of the identified object or concept. Linked Data principle also advocates use of a single data model for publishing structured data on the web—the resource description framework (RDF), a simple graph-based data model that has been designed for use in the context of the web [4]. Lastly, Linked Data uses hyperlinks to connect not only web documents but also any type of thing. For example, a hyperlink may be set between a person and a place or between a place and a company. Hyperlinks that connect things in a Linked Data context have types which describe the relationship between the things. For example, a hyperlink of the type *friend of* may be set between two people or a hyperlink of the type *based near* may be set between a person and a place. Hyperlinks in the Linked Data context are called *RDF links* in order to distinguish them from hyperlinks between classic web documents.

The RDF data model represents information as node-and-arc-labelled directed graphs. The data model is designed for the integrated representation of information that originates from multiple sources, is heterogeneously structured, and is represented using different schemata [4, 5]. Data is represented in RDF as RDF *triples*. The RDF data model is described in detail as part of the W3C RDF Primer (W3C RDF Primer, http://www.w3.org/TR/rdf-primer/).

2.3. Linked Data Mashups. Linked Data mashups are created in the similar fashion as web mashups whilst they use a unified data model, RDF model, for combining, aggregating, and transforming data from heterogeneous data resources. Using a single data model for data manipulation operations enables a simpler abstraction of application logics for mashup developers. The RDF data model is driven by vocabularies or ontologies which play the role of the common understanding among machines, developers, domain experts, and users.

A Linked Data mashup is composed from different piece of technologies. The first type of technologies is data integration which covers data transformation, storage, and accessing and application of APIs based on RDF data model. The second type of technologies is mashup execution engines which provide the execution environments for computing the mashup processing workflow. The third type of technologies is interactive programing and visualisation which provide composing and exploring environments for mashup developer to build data processing workflow for a mashup.

One simple example of a Linked Data mashup is an aggregated sales application that integrates customer relationship management (CRM) and financial data with functionality from the web and corporate backend data. This example mashup would employ real-time information, streaming content, and web services to form a coordinated application using all of these data sources. Integrated sales information for the traveling sales person could be available from their smart phone or laptop. The data integration tools are responsible for transforming streams real-time web information of financial and CRM data and background information and request for information (RFI) documents to Linked Data. Internally, internal, proprietary customer data about installed products, contracts, and upsell possibilities can be exposed as Linked Data via RDFisers [4]. When all the data are accessible as Linked Data and can be queried via SPARQL, a series of front-end applications can be built. The facet browsers for Linked Data [6, 7] enable combining financial, CRM and other data with online maps to visually identify, locate, and categorise customers for each geographical location. Using Google Maps or Mapquest (http://www.mapquest.com/) APIs, each customer site appears on the map and allows the sales person to drill down using the map paradigm to identify customer sites to expose new sales or possible upsell opportunities. Background information and RFI documents could be generated partly using semantically rich content from DBpedia (http://dbpedia.org/), the semantically structured content from Wikipedia. Integrated and updated glossary definitions of domain vernacular, references to partners and competitors could come together as competitive analysis documents. Prospective customers could read marketing evaluations combined with general reference content and links to trusted independent blogger opinions, all from a single document. Customer data can be integrated with the maps, reference information, and sales database to provide personalised content for customers.

3. Overview on Semantic Mashup Architectures

3.1. Semantic Mashup Architecture. Semantic mashup is basically a mashup application using semantic web technologies inside. Therefore, the logical architecture of semantic mashups applications is similar to the traditional mashup ones. With semantic mashups, in addition to information retrieval techniques of Web 2.0, the application architecture focuses on semantic data sources and supports techniques for sematic data preparation and processing effectively with taking out of advantages of semantic web technologies. Based on mashups execution places, it is divided into two main architectures: client-side semantic mashup application and server-side semantic mashup applications [8, 9].

3.2. Client-Side Semantic Mashup. In this architecture, the integration process of data and services is performed directly at client applications which could be browsers or smart phones.

As depicted in Figure 1, the client sends a request directly to a remote mashup service without any proxy server. The remote mashup service processes the request and response data in the requested format. Returned data is loaded in a dynamic script <*script*> in client browser.

3.3. Server-Side Semantic Mashup. With this architecture (Figure 2), services and data are integrated in the platform of service provider. Server-side semantic mashup architecture is considered a proxy-style mashup, a server component running, and a proxy for mashup service. According to this schema, all requests from clients must go through proxy server, and then remote services will be revoked. These services receive requests and return data upon the requested formats back to the proxy. The proxy could cache the data for the next similar calls and send data back to clients.

3.4. Semantic Mashup Application Levels. There are three main levels for a semantic mashup application: data level, process level, and presentation level [10].

(i) *Data level*: this level is mainly about integration and aggregation of data. The main challenges of this level are considered data extraction and integration from different data sources such as web of data, web services, or community databases. These data sources could be retrieved by web services with protocols of REST, SOAP, HTTP, and XML RPC. By any means, this level must help in extraction and integration of data and then transform them to structured models (XML, RSS/ATOM, JSON, RDF, and so on) or unstructured (audio, text, and files).

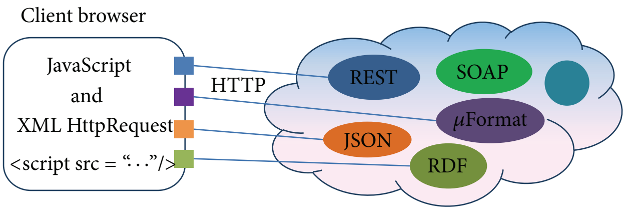


Figure 1: Client-side semantic mashup architecture.

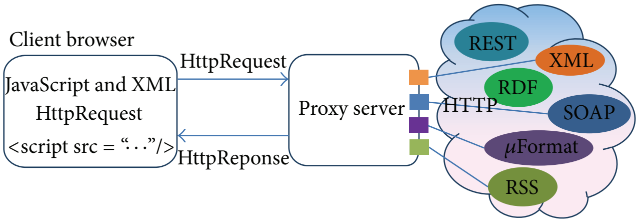


Figure 2: Server-side semantic mashup architecture.

(ii) *Process level*: at this level, a workflow for coordinating relevant web services will be defined. The integration is executed at the platform applications and coordinated workflows will be executed by associating services' APIs. In semantic mashup context, programming languages were not powerful enough for application scenario modelling. For instance, they could not provide links to different data sources or not process the interaction with browsers. Therefore, several approaches are recommended for interaction description and editing model will be discussed in the next section.

(iii) *Presentation level*: each application needs a graphical interface for user interaction, and semantic mashup is not an exception. At this level, information and semantic information processing results are presented to users. Techniques are used and ranged from an HTML page to complex applications using Ajax, JavaScript, and so on.

4. Semantic Mashups Approaches

4.1. Widget-Based Approach. With this approach, semantic mashup applications are created by visual programming tools. These tools allow developers to use predefined graphical components instead of writing codes. However, they could not provide all necessary tasks for semantic mashups, but subtasks for data extraction, merge, transform, and presentation by graphical wiring blocks [8]. They often support data sources from web services RESTful, SOAPful, RSS/ATOM, and so on. Some typical tools for this approach are JackBe Presto Wires, Microsoft Popfly, Yahoo! Pipes, Openkapow, Proto Financial, Anthracite, and Lotus Mashups.

4.2. Database-Inspired Approach. Semantic mashup applications within this approach provide common database functions such as storing data in flat files, linking, and merging data sets [11]. This approach considered the Internet as a database in which the web sources are organized as a set of tables, and each semantic mashup is a query over these tables. Hence, a mashup query language is proposed and allowed to be executed when needed. Some popular approaches in this line are YQL (Yahoo Query Language. http://developer.yahoo.com/yql/), MashQL [9], and hay MQL with Metaweb- Freebase [12].

4.3. Spreadsheet-Like Approach. In this approach, data is stored by web services and inserted directly into a spreadsheet. This means that output of a data source will be written into a given cell by users. Data values in cells are used for input parameters of another web service over the spreadsheet. There are three reasons for this mashup application.

(i) Firstly, spreadsheet is used by millions of people. It is supposed that the idea of a spreadsheet-based mashups would be widely accepted.

(ii) Secondly, spreadsheet is considered a management tool for visual reporting and data analysis with simple functions. This is suitable with simple tasks of a mashup application: reusing and linking data from different sources.

(iii) Thirdly, spreadsheet is a favourable environment for end users as follows: it (1) provides a real-time execution environment, (2) supports step-by-step tasks, and (3) provides high flexibility as its cell-based relationship.

4.4. Visual Scripting Language Approach. This approach includes mashup applications developed based on script languages such as Google Mashup Editor (GME), Web Mashup Scripting Language (WMSL), and WSO2 Mashup Server [13]. These tools are useful for users in building simple mashup applications with client-side or server-side architecture. Hence, this approach is difficult for user in developing complex mashup scenarios.

4.5. Automatic Creation Approach. A mashup generation with automatic creation has been developed recently, and it has attracted the community research. A mashup includes smaller components called mashlets for executing specific functions [14]. For instance, a mashlet is for modelling RSS data sources, drawing a map, or data extraction from a RSS input source. Mashlet could be at logical level for combining different mashlets together, so-called glue pattern (GP). For example, a GP can combine three mashlets above to draw a map from RSS data sources.

With this idea, mashup building will be the selection of typical or specific mashlets and identification of GP for combining them. In order to support this, some tools have been developed for support users in a way called *autocompletion*. The underneath idea is about the simplicity and visualisation: users first select initial mashlets for their mashups-to-be;

the system then recommends GP possibilities and relevant mashlets based on the mashlet repository and a *collective wisdom* from experiences of successful usages before. Popular tools for this approach are MaxMash and MatchUp.

4.6. Semantic Pipes Approach. Based on the abstract concept of *pipe* and its typical features in processing pipelines of the core design of UNIX systems, there are researches such as Yahoo! Pipes (http://pipes.yahoo.com/pipes/) that proposed a way of information integration on the distributed web environment in order to create information mashups application for easily information integration and reuse by pipes connection. This means each pipe is an input data source of another one. Specifically, each pipe is considered a task for common query processing including a set of subqueries coordinated by the user and unified to a common data format through predefined pip operators.

With Yahoo! Pipes, this is an application support creating mashup application based on the customisation of services and information flows with simple operators and traditional control structures (loop, case of, and so on). However, RDF is not supported in this approach.

With DERI Pipes [15], the approach focuses on the RDF data sources and supports users making semantic web applications with RDF. The concept of semantic web pipe (SWP) in this framework enables crate and reuse components for semantic data processing. A *semantic web pipe* is a predefined workflow that, given a set of RDF sources (resolvable URIs), composes and processes them by means of pipelined special purpose operators. Semantic web pipes (SWP) do not aim at replacing complex workflow languages rather than promoting a very reduced acyclic data processing model. More specifically, it only supports two of workflow patterns of split and merge [15]. Besides, SWP provides typical operators and a web-based graphical environment to support users in data extraction and processing according to semantic web standards. Hence, this approach supports creating semantic data processing for semantic mashup applications.

5. Technologies Enabling Linked Data Mashups

5.1. Data Integration. Data integration technologies for Linked Data mashups involve all solutions and tools to enable data from heterogeneous sources accessible as Linked Data. The representative architecture for data integration of Linked Data mashups is depicted in Figure 3.

In this architecture, the publishing layer provides all tools to expose traditional data sources in RDF data formats. They include wrappers for the databases and RDFizers for transforming data from other formats (e.g., XML, JSON, and HTML) into RDF. Then, when all data is accessible as Linked Data, it might be stored in storages or accessed via Web APIs such as SPARQL endpoints, called web Linked Data. These data might be manipulated and integrated to access in a refined form via a SPARQL query interface by application code in the application layer.

5.2. Mashup Execution Engines. A mashup is usually constructed in a formal language to represent the computing process that generates the output for the mashup. Then, the mashup represented in such language is executed in an execution engine. In this section, we introduce two popular execution engines, MashMaker [3] and DERI Pipes [15]. MashMaker uses functional programing language whilst DERI Pipes uses Domain Specific Language (DSL) in XML.

MashMaker provides a modern functional programming language with nonside affecting expressions, higher order functions, and lazy evaluation. MashMaker programs can be manipulated either textually or through an interactive tree representation, in which a program is presented together with the values it produces. MashMaker expressions are evaluated lazily. The current consensus in the programming language community seems to be the lazy evaluation that is wrong for conventional programming languages. This is because the bookkeeping overhead of lazy evaluation makes programs run slowly, the complex evaluation behaviour makes performance hard to predict, and programmers often have to battle with space leaks due to long chains of lazy thunks. In the case of web mashups, the bookkeeping cost of remembering how to evaluate something is tiny compared to the massive cost of fetching and scraping a web site; thus it is only necessary for a very small number of expressions to be unneeded for the bookkeeping cost to be more than paid back. Even if fetching a web site was cheap, it is important for us to minimize the number of queries we make to a remote server, to avoid overwhelming a server. Typical mashup programs work with relatively small amounts of data that are not directly presented to the user, and so space leaks are far less of a problem.

DERI Pipes [15] propose a flexible architectural style for the fast development of reliable data intensive applications using RDF data. Architectural styles have been around for several decades and have been the subject of intensive research in other domains such as software engineering and databases. Le-Phuoc et al. [15] base their work on the classical pipe abstraction and extend it to meet the requirements of (semantic) web applications using RDF. The pipe concept lends itself naturally to the data-intensive tasks at hand by its intrinsic concept of decomposing an overall data-integration and processing task into a set of smaller steps which can be freely combined. This resembles a lot the decomposition of queries into smaller subqueries when optimizing and generating query plans. To some extent, pipes can be seen as materialized query plans defined by the application developer. Besides the intrinsic encapsulation of core functionalities into small components, this paradigm is inherently well suited to parallel processing which is an additional benefit for high-throughput applications which can be put on parallel architectures.

5.3. Interactive and Visual Programming. As more and more reusable structured data appears on the web, casual users will want to take into their own hands the task of mashing up data rather than waiting for mashup sites to be built that address exactly their individually unique needs. Therefore,

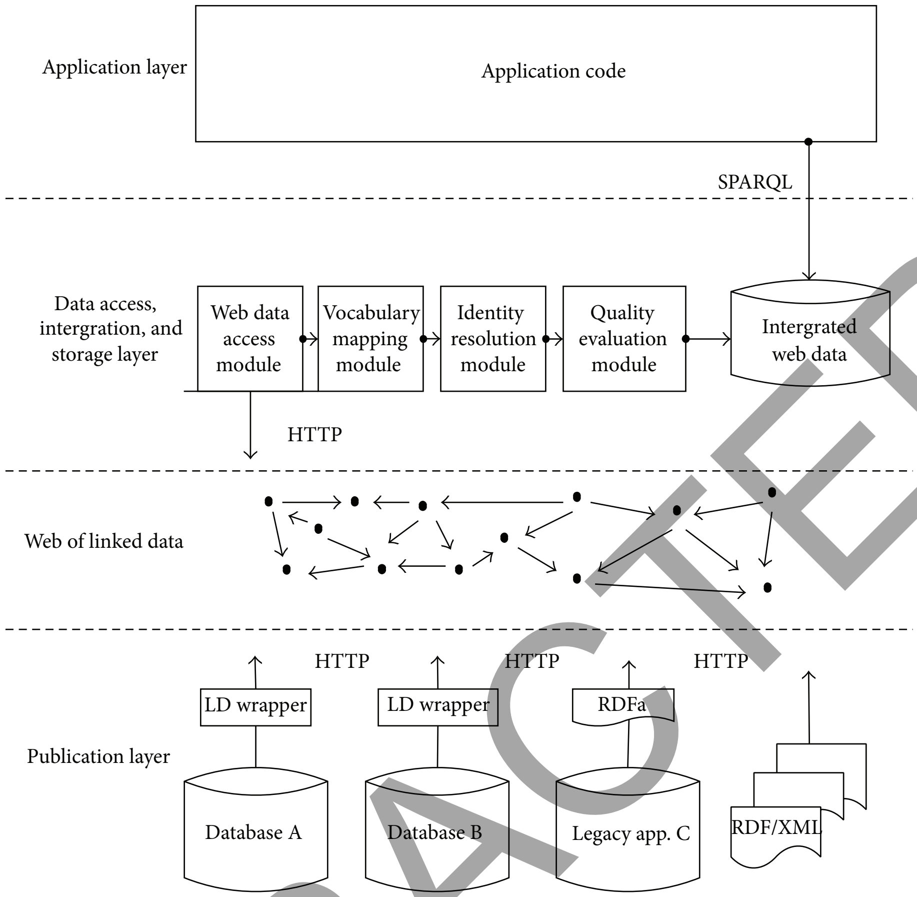


Figure 3: Data integration architecture for Linked Data mashups [4].

an interactive and visual programming environment is desired for building Linked Data mashups. The techniques and tools like facet-browsing and Web GUI facilitate interactive mashup developing editors such as Potluck [7], Exhibit [6], and IntelMash Maker [3].

Potluck [7] provides a web user interface that lets casual users—those without programming skills and data modelling expertise—mash up data themselves. Potluck is novel in its use of drag and drop for merging fields, its integration and extension of the faceted browsing paradigm for focusing on subsets of data to align, and its application of simultaneous editing for cleaning up data syntactically. Potluck also lets the user construct rich visualizations of data in-place as the user aligns and cleans up the data. This iterative process of integrating the data while constructing useful visualizations is desirable when the user is unfamiliar with the data at the beginning—a common case—and wishes to get immediate value out of the data without having to spend the overhead of completely and perfectly integrating the data first.

Exhibit [6] is a lightweight framework for publishing structured data on standard web servers that requires no installation, database administration, or programming. Exhibit enables authors with relatively limited skills to publish richly interactive pages that exploit the structure of their data for better browsing and visualization. Such structured publishing in turn makes that data more useful to all of its consumers: individual readers get more powerful interfaces, mashup creators can more easily repurpose the data, and semantic web enthusiasts can feed the data to the nascent semantic web.

IntelMashMaker [3] does this by making mashup creation part of the normal browsing process. Instead of having a reasonably skilled user who can create a mashup in advance as a mashup site, where other users browse, MashMaker creates personalized mashups for the user inside their web browser. Rather than requiring the fact that a user tells a mashup tool what they want to create, MashMaker instead watches what information the user looks at, correlates the user's behaviour with that of other users, and guesses a mashup application that the user would find useful, without the user even having to realize what they wanted for a mashup.

5.4. DBpedia Mashups. If you see Wikipedia as a main place where the knowledge of mankind is concentrated, then DBpedia—which is extracted from Wikipedia—is the best place to find the machine representation of that knowledge [5]. DBpedia constitutes a major part of the semantic data on the web. Its sheer size and wide coverage enable you to use it in many kinds of mashups: it contains biographical, geographical, bibliographical data, as well as discographies, movie metadata, technical specifications, and links to social media profiles and much more. Just like Wikipedia, DBpedia is a truly cross-language effort; for example, it provides descriptions and other information in various languages. DBpedia is an unavoidable resource for applications dealing with commonly known entities like notable persons, places, and for others looking for a rich hub connecting other semantic resources.

5.5. Mashups for Internet of Things. Internet of things (IoT) has been creating vast amount of distributed stream data which can be modelled using RDF data model called Linked Stream Data. Linked Stream Data is becoming new valuable data sources for Linked Data mashups. Therefore, the web of things (WoT) together with mashup-like applications is gaining popularity with the development of the Internet towards a network of interconnected objects, ranging from cars and transportation cargos to electrical appliances.

A long the same line, cities are alive: they rise, grow, and evolve like living beings; WoT allows a wide range of smart city applications. In essence, the state of a city changes continuously, influenced by a lot of factors, both human (people moving in the city or extending it) and natural ones (rain or climate changes). Cities are potentially huge sources of data of any kind and for the last years a lot of efforts have been put in order to create and extract those sources. This scenario offers a lot of opportunities for mashup developers: by combining and processing the huge amount of data (both public and private), it is possible to create new services for urban stakeholders—citizens, tourists, and so forth, called urban mashups [16].

Another application domain for IoT is emergency management [17]. Emergency management applications support a command staff in disruptive disaster situations, such as earthquakes, large-scale flooding, or fires. One crucial requirement to emergency management systems is to provide decision makers with the relevant information to support their decisions. Mashups can help here by providing flexible and easily understandable views on up-to-date information.

5.6. Tourism Mashups. Web 2.0 has revolutionized the way users interact with information, by adding a vast amount of services, where end users explicitly and implicitly, and as a side effect of their use, generate content that feeds back into optimization of these services. The resulting (integrated) platforms support users in and across different facets of life, including discovery and exploration and travel and tourism. Linked Data mashup enables the creation and use of travel mashups, defined based on the varied travel information needs of different end users, spanning temporal, social, and spatial dimensions [18]. The RDF-based travel mashups are created for bridging these dimensions, through the definition and use of composite, web-, and mobile-based services. Their applications elicit the information need of an end user exploring an unfamiliar location and demonstrate how the Topica Travel Mashup leverages social streams to provide a topical profile of points of interest that satisfies these user's requirements.

5.7. Biological and Life Science Domains. Semantic web technologies provide a valid framework for building mashups in the life sciences. Ontology-driven integration represents a flexible, sustainable, and extensible solution to the integration of large volumes of information. Additional resources, which enable the creation of mappings between information sources, are required to compensate for heterogeneity across namespaces. For instance, [19] uses an ontology-driven approach to integrate two gene resources (Entrez Gene and HomoloGene) and three pathway resources (KEGG, Reactome, and BioCyc), for five organisms, including humans. Sahoo et al. [19] created the Entrez Knowledge Model (EKoM), an information model in OWL for the gene resources, and integrated it with the extant BioPAX ontology designed for pathway resources. The integrated schema is populated with data from the pathway resources, publicly available in BioPAX-compatible format, and gene resources for which a population procedure was created. The SPARQL query language is used to formulate queries over the integrated knowledge base to answer the three biological queries. Simple SPARQL queries could easily identify hub genes, that is, those genes whose gene products participate in many pathways or interact with many other gene products. The identification of the genes expressed in the brain turned out to be more difficult, due to the lack of a common identification scheme for proteins.

6. Open Challenges

Even though there have been a plenty of technology and research achievements of Linked Data community to enable Linked Data mashups, there are a number of challenges to address when building mashups from different sources. The challenges can be classified into four groups: entity extraction from text, object identification and consolidation, abstraction level mismatch, and data quality.

Transforming Text Data to Symbolic Data for Linked Data Entities. A large portion of data is described in text. Human language is often ambiguous—the same company might be referred to in several variations (e.g., IBM, International Business Machines, and Big Blue). The ambiguity makes cross-linking with structured data difficult. In addition, data expressed in human language is difficult to process via software programs. Hence overcoming the mismatch between documents and data to extract RDF-based entities is still emerging challenges.

Object Identification and Consolidation. Structured data are available in a plethora of formats. Lifting the data to a common data format is thus the first step. But even if all data is available in a common format, in practice, sources differ in how they state what essentially the same fact is. The differences exist both on the level of individual objects and on the schema level. As an example for a mismatch on the object level, consider the following: the SEC uses a so-called Central Index Key (CIK) to identify people (Chief Executive Officers and Chief Financial Officers), companies, and financial instruments while other sources, such as DBpedia, use URIs to identify entities. In addition, each source typically uses its own schema and idiosyncrasies for stating what essentially the same fact is. Thus, methods have to be in place for reconciling different representations of objects and schemata.

Abstraction Levels. Data sources provide data at incompatible levels of abstraction or classify their data according to taxonomies pertinent to a certain sector. Since data is being published at different levels of abstraction (e.g., person, company, country, or sector), the data aggregated for the individual viewpoint may not be matched to the data (e.g., from statistical offices). Also, there are differences in geographic aggregation (e.g., region data from one source and country-level data from another). A related issue is the use of local currencies (USD versus EUR) which have to be reconciled in order to make data from disparate sources comparable and amenable for analysis.

Data Quality. Data quality is a general challenge when automatically integrating data from autonomous sources. In an open environment the data aggregator has little to no influence on the data publisher. Data is often erroneous, and combining data often aggravates the problem. Especially when performing reasoning (automatically inferring new data from existing data), erroneous data has potentially devastating impact on the overall quality of the resulting dataset. Hence, a challenge is how data publishers can coordinate in order to fix problems in the data or blacklist sites which do not provide reliable data. Methods and techniques are needed to check integrity and accuracy, also highlight, identify, and check sanity, corroborating evidence, and assess the probability that a given statement is true and equate weight differences between market sectors or companies, and act as clearing houses for raising and settling disputes between competing (and possibly conflicting) data providers and interact with messy erroneous web data of potentially dubious provenance and quality. In summary, errors in signage, amounts, labelling, and classification can seriously impede the utility of systems operating over such data.

7. Case Study: Semantic Mashup with Semantic Pipes and MetaWeb

7.1. Semantic Pipes for Information Integration. A *Semantic web pipe* is a predefined workflow that, given a set of RDF sources (resolvable URIs), composes and processes them by means of pipelined special purpose operators. Semantic web pipes (SWP) do not aim at replacing complex workflow languages rather than promoting a very reduced acyclic data processing model. More specifically, it only supports two of workflow patterns of split and merge [15]. A *pipe* is a set of instances of the operators.

(1) Each fixed parameter input and all variable arty parameter inputs are linked to either (i) quoted literals such as “`<?xml version = "1.0" ? > ...`” as fixed string input, (ii) a URI such as `<http://alice.example.org>` denoting a web retrievable data source that contains data in the required input format, or (iii) the output of another pipe.

(2) All but one output (the “overall” output of the pipe) are linked inputs of other operators.

(3) Links between inputs and outputs are acyclic.

By following these constraints, each pipe can itself be used as an operator in other pipes. Default output data format for semantic web pipes operators are RDF. Furthermore, it also supports other formats such as RDF/JSON and RDF/XML or other RDF formats Turtle/N3, TRIG, TRIX, and N-TRIPLES in order to ensure the flexibility with the pipes [15]. Semantic web pipes architecture is described in (Figure 4). As depicted in Figure 4, the mashup engine interacts with SWP based on output data with formats complying with semantic web.

7.2. Data Sources for Semantic Mashups. Freebase [20], a product of Metaweb (now under Google), is a community-curated database of well-known people, places, and things. Freebase is a graph database of connected subgraphs and becomes open and commonly used data storage for specific domains. Freebase also provides a query language, namely, MQL (Metaweb Query Language) [12] for data retrieval over HTTP and output data format is serialised in JSON. Besides, with support of domain specific data modelling, Freebase enables creating open data community about deferent domains. This helps mashups applications easily exchange data of concerned application domains.

As depicted in Figure 5, Freebase graphs use blue balls to represent graph nodes and arrows connecting balls are relations (properties) between those nodes. Each relation is labelled with a property identifying the type of the relation. For example, given a sample data as *Britney Spears* singer created “*Baby one more time*” album in which there is a song “*Sometimes*.”

7.3. Semantic Mashups-Based Information Integration with Semantic Web Pipes and Freebase. Tuchinda et al. [14] defined five tasks for a semantic mashups application: data retrieval, data cleaning, source modelling, data integration, and data visualization. Raw data will be retrieved, cleaned and represented in a common schema. These data will be then linked and integrated according a given mashups scenario. Finally, the mashup data will be published for users. Five mentioned tasks are background ones belong to three basic levels of a semantic mashup application as described in Figure 6.

As described in Figure 6, Freebase plays the role of an open data for a specific domain for data retrieval task. SWP is the vital tool for modeling the retrieved data and integrating these data by using the typical operators.

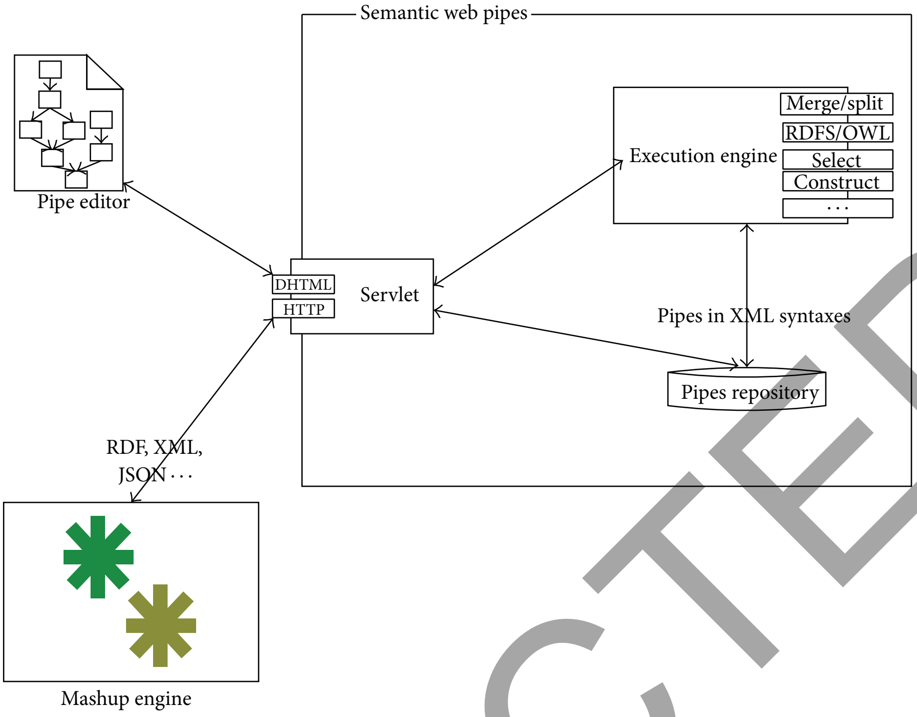


Figure 4: Semantic web pipes architecture [15].

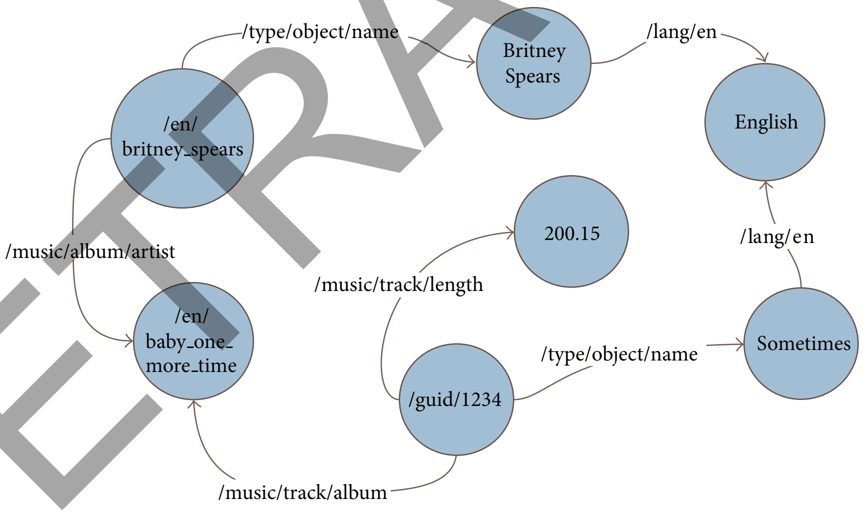


Figure 5: A sample of data in Freebase.

7.4. Semusic: Music Semantic Mashups. Semusic is built as a web-based semantic mashup application enabling users search for information in musical domain. Semusic is developed with its database of music information from Freebase with integration with different relevant data sources. The integration is implemented by SWP operators and inherited Metaweb services in order to provide relevant information to search keywords. The integrated data is modelled with music ontology with objects identified by Metaweb. Semusic aims at integrating information of four objects: music genre, music artists, music albums, and music tracks. Semusic supports searching those objects with information integrated from open datasets of music such as MusicBrainz, Wikipedia, and Tinysong.

7.4.1. Semusic Logical Structure. As presented in Figure 7, *Semusic* is designed in separate classes which are grouped into 3 packages: end-user interface, data/service integration and web service APIs, and web data sources. With this

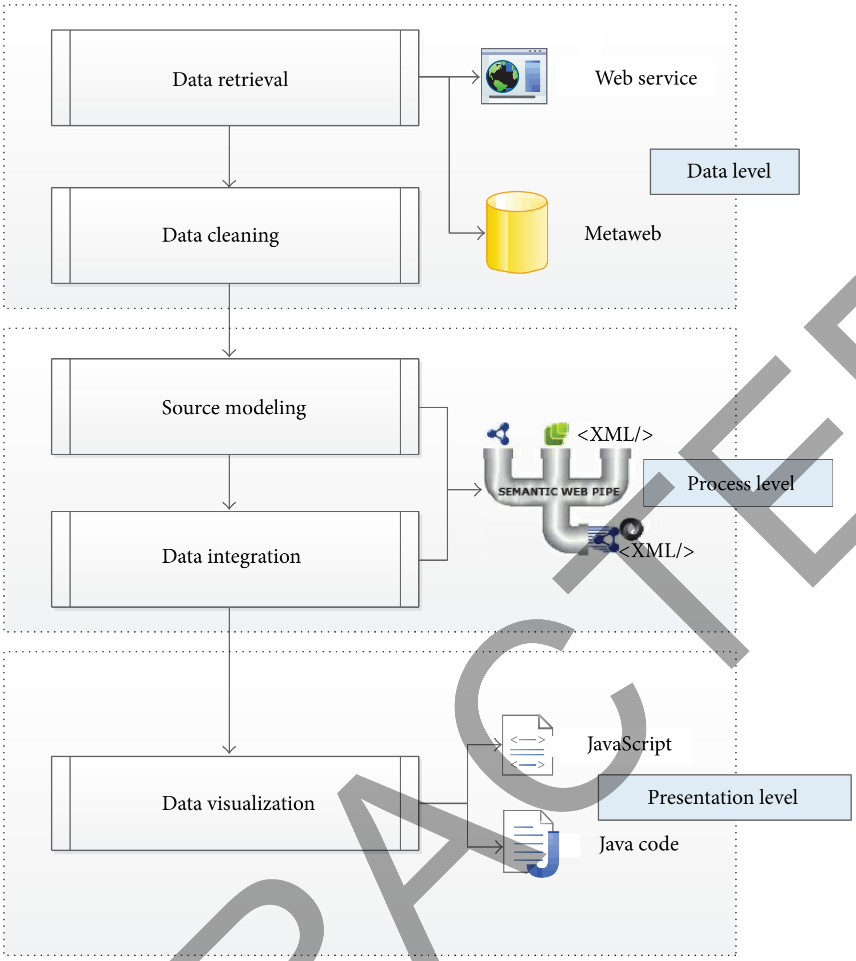


Figure 6: Semantic mashup application structure with SWP and Freebase (Metaweb).

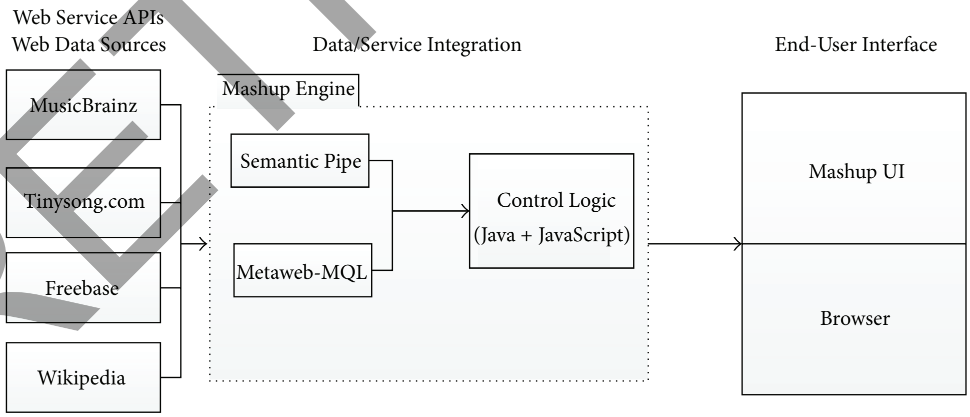


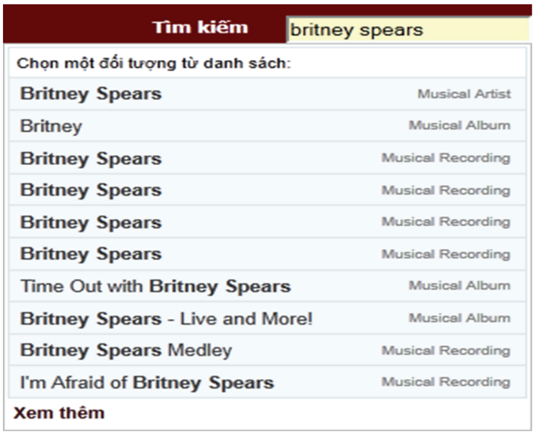
Figure 7: Semusic logical structure.

structure, Semusic complies with the mentioned architecture of a semantic mashup application depicted in Figure 6.

7.4.2. Implementation and Evaluation. With the objective of building Semusic as a prototype demonstrating for information integration using semantic mashups for data sources and web services in music domain, Semusic implementation includes following features: keyword-based search and information presentation.

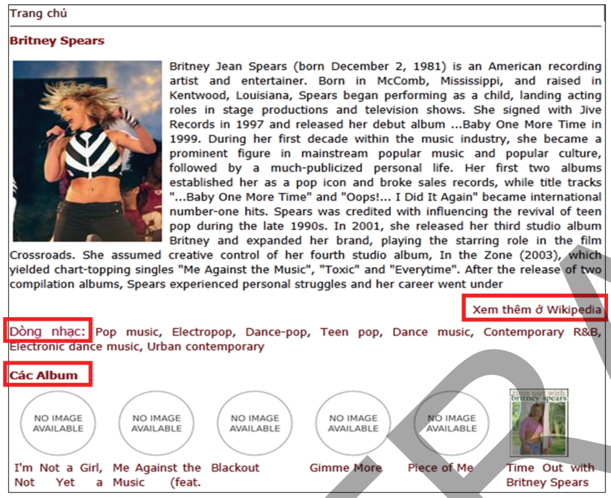
The search feature will seek for objects based on user's keywords. The system will recommend specific objects related to the searching keywords as depicted in Figure 8.

The graphical user interface of search results presentation (Figure 9) shows descriptions of found resources with



Tìm kiếm	britney spears
Chọn một đối tượng từ danh sách:	
Britney Spears	Musical Artist
Britney	Musical Album
Britney Spears	Musical Recording
Britney Spears	Musical Recording
Britney Spears	Musical Recording
Britney Spears	Musical Recording
Time Out with Britney Spears	Musical Album
Britney Spears - Live and More!	Musical Album
Britney Spears Medley	Musical Recording
I'm Afraid of Britney Spears	Musical Recording
Xem thêm	

Figure 8: Keyword-based search with Semusic.



Trang chủ

Britney Spears

Britney Jean Spears (born December 2, 1981) is an American recording artist and entertainer. Born in McComb, Mississippi, and raised in Kentwood, Louisiana, Spears began performing as a child, landing acting roles in stage productions and television shows. She signed with Jive Records in 1997 and released her debut album ...Baby One More Time in 1999. During her first decade within the music industry, she became a prominent figure in mainstream popular music and popular culture, followed by a much-publicized personal life. Her first two albums established her as a pop icon and broke sales records, while title tracks "...Baby One More Time" and "Oops!... I Did It Again" became international number-one hits. Spears was credited with influencing the revival of teen pop during the late 1990s. In 2001, she released her third studio album Britney and expanded her brand, playing the starring role in the film Crossroads. She assumed creative control of her fourth studio album, In the Zone (2003), which yielded chart-topping singles "Me Against the Music", "Toxic" and "Everytime". After the release of two compilation albums, Spears experienced personal struggles and her career went under

Xem thêm ở Wikipedia

Dòng nhạc: Pop music, Electropop, Dance-pop, Teen pop, Dance music, Contemporary R&B, Electronic dance music, Urban contemporary

Các Album

I'm Not a Girl, Not Yet a

Me Against the Music (feat.

Blackout

Gimme More

Piece of Me

Time Out with Britney Spears

Figure 9: Information presentation in Semusic.

relevant information from and relevant information at the main page (music genre, albums, etc.).

There are more functions of Semusic also have efficiently developed based on the structure mentioned above. However, the main challenge is that the open data sources are not complete and not well updated as well as the modelling languages of those data sources. One suggestion is about using a unified data format as Linked Data based-on RDF.

8. Conclusion and Outlook

The evolution of data sources on the web of data has made a strong wave of research approaches in semantic mashups. In this paper we have investigated state-of-the-art approaches in Linked Data mashups in terms of technologies and application domain. From analytical reviews on approaches, we have drawn up open challenges for Linked Data mashups. This review is a first step of our research aiming at pointing out research trends in building up real applications. They are based on the open linked data in order to make a new leash of intelligent applications that utilise and facilitate the advantages of RDF data model and Linked Data for new generation of Linked Data mashups application line.

Besides, tools for building semantic mashup applications have been also developed [21, 22]. Those applications have not reused the already made mashups. Based on this remark, it can be helpful to develop complex systems by reusing and inheriting from built mashup ones. The SWP architecture makes us an enabling environment for accomplishing this task. However, SWP currently only supports fetching data from different open data sources rather than data sources from web services. Therefore, this paper proposed an approach of semantic mashups based on SWP and Metaweb-Freebase, and this is an efficient way for practical semantic mashup applications.

For the future work, we would like to focus on following issues: (i) defining new SWP operators for identifying semantic web services and fetching data from semantic web services and (ii) modelling users for user ontologies to help users define and utilise data sources effectively.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was sponsored by Vietnam's National Foundation for Science and Technology Development (NAFOSTED) in the framework of the Grant no. 102.02-2010.14. This work was supported by the Yeungnam University research grants in 2011. Also, it was partially supported by the BK21+ of the National Research Foundation of Korea.

References

[1] T. Berners-Lee, "Linked Data—Design Issues," W3C, 2006.

[2] B. Endres-Niggemeyer, *Semantic Mashups*, Springer, Berlin, Germany, 2013.

[3] R. Ennals, E. Brewer, M. Garofalakis, M. Shadle, and P. Gandhi, "Intel mash maker: join the web," *ACM SIGMOD Record*, vol. 36, no. 4, pp. 27–33, 2007.

[4] T. Heath and C. Bizer, *Linked Data: Evolving the Web into a Global Data Space*, Morgan & Claypool, 2011.

[5] M. Héder and I. Solt, "DBpedia mashups," in *Semantic Mashups*, B. Endres-Niggemeyer, Ed., pp. 119–143, Springer, Berlin, Germany, 2013.

[6] D. F. Huynh, D. R. Karger, and R. C. Miller, "Exhibit: lightweight structured data publishing," in *Proceedings of the 16th International World Wide Web Conference (WWW '07)*, pp. 737–746, ACM, May 2007.

[7] D. F. Huynh, R. C. Miller, and D. R. Karger, "Potluck: data mash-up tool for casual users," *Journal of Web Semantics*, vol. 6, no. 4, pp. 274–282, 2008.

[8] E. Ort, S. Brydon, and M. Basler, "Mashup styles, part 1: server-side mashups," Sun Technical Article, 2007.

[9] E. Ort, S. Brydon, and M. Basler, "Mashup styles, part 2: client-side mashups," Sun Technical Article, 2007.

[10] E. M. Maximilien, H. Wilkinson, N. Desai, and S. Tai, "A domain-specific language for web APIs and services mashups," in *Proceedings of the International Conference on Service-Oriented Computing (ICSOC '07)*, B. Krämer, K. J. Lin, and P. Narasimhan, Eds., vol. 4749 of *Lecture Notes in Computer Science*, pp. 13–26, Springer, Berlin, Germany.

[11] M. Jarrar and M. D. Dikaiakos, "A data mashup language for the data web," in *Proceedings of the Workshop Linked Data on the Web (LDOW '09)*, 2009.

[12] "GoogleDevelopers: MQL Overview," 2014.

[13] O. Hartig and A. Langegger, "A database perspective on consuming linked data on the web," *Datenbank Spektrum*, vol. 10, pp. 57–66, 2010.

[14] R. Tuchinda, P. Szekely, and C. A. Knoblock, "Building Mashups by example," in *Proceedings of the 13th International Conference on Intelligent User Interfaces (IUI '08)*, pp. 139–148, ACM, January 2008.

[15] D. Le-Phuoc, A. Polleres, M. Hauswirth, G. Tummarello, and C. Morbidoni, "Rapid prototyping of semantic mash-ups through semantic web pipes," in *Proceedings of the 18th International World Wide Web Conference*, pp. 581–590, ACM, 2009.

[16] D. Dell'Aglio, I. Celino, and E. Della Valle, "Urban mashups," in *Semantic Mashups*, B. Endres-Niggemeyer, Ed., pp. 287–319, Springer, Berlin, Germany, 2013.

[17] A. Schulz and H. Paulheim, "Mashups for the emergency management domain," in *Semantic Mashups*, B. Endres-Niggemeyer, Ed., pp. 237–260, Springer, Berlin, Germany, 2013.

[18] A. Cano, A. S. Dadzie, and F. Ciravegna, "Travel mashups," in *Semantic Mashups*, B. Endres-Niggemeyer, Ed., pp. 321–347, Springer, Berlin, Germany, 2013.

[19] S. S. Sahoo, O. Bodenreider, J. L. Rutter, K. J. Skinner, and A. P. Sheth, "An ontology-driven semantic mashup of gene and biological pathway information: application to the domain of nicotine dependence," *Journal of Biomedical Informatics*, vol. 41, no. 5, pp. 752–765, 2008.

[20] "Google: Freebase," 2014, http://www.freebase.com/.

[21] J. J. Jung, "ContextGrid: a contextual mashup-based collaborative browsing system," *Information Systems Frontiers*, vol. 14, no. 4, pp. 953–961, 2012.

[22] J. J. Jung, "Collaborative browsing system based on semantic mashup with open APIs," *Expert Systems with Applications*, vol. 39, no. 8, pp. 6897–6902, 2012.