
Semantic web rules and ontologies for developing personalised mashups

Aikaterini K. Kalou and
Dimitrios A. Koutsomitropoulos*

High Performance Information Systems Laboratory (HPCLab),
Computer Engineering and Informatics Department,
School of Engineering, University of Patras,
Building B, 26500 Patras-Rio, Greece
E-mail: kaloukat@hpclab.ceid.upatras.gr
E-mail: kotsomit@ceid.upatras.gr

*Corresponding author

Theodore S. Papatheodorou

Deceased; formerly of: High Performance Information
Systems Laboratory (HPCLab),
Computer Engineering and Informatics Department,
School of Engineering, University of Patras,
Building B, 26500 Patras-Rio, Greece

Abstract: The current trends for the future evolution of the web are without doubt the Semantic Web and Web 2.0. A common perception for these two visions is that they are competing. Nevertheless, it becomes more and more obvious that these two concepts are complementary. Semantic web technologies have been considered as a bridge for the technological evolution from Web 2.0 to Web 3.0, the web about recommendation and personalisation. Towards this perspective, in this work, we introduce a framework based on a three-tier architecture that illustrates the potential for combining Web 2.0 mashups and Semantic Web technologies. Based on this framework, we present an application for searching books from Amazon and Half eBay with a focus on personalisation. This implementation purely depends on ontology development, writing of rules (for the personalisation), and on creation of a mashup with the aid of web APIs. However, there are several open issues that must be addressed before such applications can become commonplace. The aim of this work is to be a step towards supporting the development of applications which combine the two trends so as to conduce to the term Web 3.0, which is used to describe the next generation web.

Keywords: Semantic Web; knowledge representation; ontology; rules; SWRL; personalisation; Web 2.0; mashups; web APIs; Web 3.0; 3-tier architecture; linked data.

Reference to this paper should be made as follows: Kalou, A.K., Koutsomitropoulos, D.A. and Papatheodorou, T.S. (2013) 'Semantic web rules and ontologies for developing personalised mashups', *Int. J. Knowledge and Web Intelligence*, Vol. 4, Nos. 2/3, pp.142–165.

Biographical notes: Aikaterini K. Kalou is a researcher at the High Performance Information Systems Laboratory (HPCLab), University of Patras. She is currently a PhD candidate at the Computer Engineering and Informatics Department from which she has received an MSc and a Computer and Informatics Engineering Diploma. Her research interests include ontological engineering, web engineering, semantic mashups, web services, personalisation techniques, linked open data and the Semantic Web.

Dimitrios A. Koutsomitropoulos is an Adjunct Assistant Professor at the Department of Computer Engineering and Informatics and a Research Fellow at the High Performance Information Systems Laboratory (HPCLab), University of Patras. He has received a fellowship-awarded PhD and an MSc from the same department, specialising in knowledge management and discovery on the web. For over ten years he has lead and participated in many R&D projects with European and national funding. His research interests include knowledge discovery, ontological engineering, semantic interoperability and Semantic Web, where he has published an important number of research articles and papers.

Theodore S. Papatheodorou was a Professor at the Department of Computer Engineering and Informatics, University of Patras, from 1984 to 2012. He served as the Department's Chair between 2005 and 2009. He was the Head of the High Performance Information Systems Laboratory (HPCLab). He received his PhD in Computer Science in 1973 and his MSc in Mathematics in 1971 from Purdue University as well as a BSc in Mathematics from University of Athens in 1968. He authored hundreds of scientific publications in several areas of computer engineering and computer science.

1 Introduction

The Semantic Web and Web 2.0 are two seemingly competing visions that dominate in web research and development. It is our firm belief that the technologies and the core strengths of these visions are complementary, rather than in competition. In fact, both technologies need each other in order to scale beyond their own drawbacks. And this, in a way that enables forthcoming web applications to combine Web 2.0 principles, especially those that set off notions such as usability, community and collaboration, with the powerful Semantic Web infrastructure, which facilitates the information sharing among web applications.

By adding the Semantic Web to Web 2.0, we move conceptually closer to Web 3.0. The underlying technologies of the Semantic Web, which enrich content, and the intelligence of the social web, pull in user profiles and identities, and must be combined for Web 3.0 to work (Cho, 2008). Consequently, the incorporation of Semantic Web and Web 2.0 principles will conduce to the development of Web 3.0, the web about personalisation and recommendation.

Towards this direction, in this work we attempt to build a web application based on a 3-tier architecture (Pomonis et al., 2009), which combines basic principles of Semantic Web with Web 2.0 mashups. The implementation of this application, named *Books@HPCLab*, is focused mainly on the ontology development and on the mashup creation, since ontologies and mashups are the pillars of the Semantic Web and Web 2.0,

respectively. A main characteristic of the application – and at the same time the primary benefit of Web 3.0 – is users' personalisation, which is implemented with the use of *rules*.

Users have the ability to search and find metadata and sell-offers for books, which fit their personal preferences. These data are in fact fetched and combined from different and heterogeneous sources on the web, like Amazon and Half eBay, thus forming a data *mashup*. Book metadata are finally triplified, linked to their offers and kept in an ontology (*BookShop ontology*). This makes the application's presented information more reusable and effectively more sharable. At the same time, this allows us to easily open up this information to the *linked and open data (LOD)* world, by providing a corresponding interface.

To make an assessment of our system's efficiency and performance we conduct an evaluation experiment, which involves real users posing queries from a predefined query set. Captured user behaviour and system response are then used to compute the re-ranking efficiency of the system as well as some quantitative and performance metrics, which consider the semantic and ontological aspect of its design.

The following text is organised in nine sections. In Section 2, we start by providing some broad definitions and discussing the concepts of Semantic Web, Web 2.0, Web 3.0 and linked data, putting a special focus on personalisation based on semantic technologies. At the same time, we identify some major design decisions in our implementation. Furthermore, in Section 3, we discuss related work and the theoretical background of the research area. In Section 4, we describe in detail our application, its components, its architecture, the developed ontology and the personalisation rules. In Section 5, we explain step-by-step the entire process of collecting data from online bookstores. Next, Section 6 outlines some indicative application scenarios in order to illustrate the features and the functionality of the application. Section 7 includes the methodology and results of the evaluation experiment. Based on our implementation, in Section 8, we discuss and advise about some key issues that are to be met in similar ventures and identify potential future directions. Finally, Section 9 summarises our conclusions.

2 Background

Nowadays, the term Web 3.0 is used to describe the evolvement of the web for the next decade (2010 to 2020) (Lassila and Handler, 2007). Web 3.0 will surely incorporate Semantic Web notions and Web 2.0 principles, but it will also include some more sophisticated concepts like artificial intelligence, as other researchers believe.

2.1 Semantic web

The Semantic Web is not a separate web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation. In this context, web content is presented in a form that is more easily machine-understandable, which means that machines will become much better able to process, to 'understand' and to integrate the information that they simply display at present (Berners-Lee et al., 2001).

Ontologies are without doubt the ‘backbone’ of the Semantic Web. The most simple and easily understandable definition of the term *ontology* is proposed by W3C: “An ontology formally defines a common set of terms that are used to describe and represent the basic concepts in a domain and the relationships among them”. A number of languages for expressing ontologies have been specified, all with different levels of granularity. In our system, we choose OWL (Bechhofer et al., 2004) for ontology representation, which extends RDF and is expressive enough for our purposes.

At the core of the Semantic Web architecture stack appears *reasoning*, the key component for the derivation of facts expressed implicitly in an ontology. Semantic reasoners and frameworks such as Pellet, FaCT++, Jena and others are pieces of software which implement the aforesaid task. In our implementation we choose Pellet which, in addition of being a native Java application, offers enhanced facilitation of reasoning operations.

Note also that many ontology languages have restrictions on their expressiveness for the sake of decidability. One way to address this problem is to extend these languages with some form of *rules*. Especially for OWL, the extension of OWL DL with Horn-like rules gives a very expressive language, named Semantic Web rule language (SWRL), which is intended to be the rule language of the Semantic Web (Antoniou et al., 2005b; Hitzler and Parsia 2009), although being undecidable in its entirety. To overcome this risk, a *safety* condition is imposed, known as ‘DL-safety’ and such rules are called ‘DL-safe SWRL rules’ and are used in the context of our work.

Linked data are also included among the means to reach the vision and accomplish the scope of the Semantic Web. In summary, the concept of linked data focuses on the creation of typed links between different data sources on the web. Linked data, from a technical aspect, are data with explicitly defined meaning, published on the web in a machine-readable format (data in RDF), which are linked to other external datasets and can also be linked to from external datasets.

2.2 Web 2.0, Web 3.0 and linked data

The main characteristic of Web 2.0 is that it provides great value to the end user web utilisation, by promoting notions such as interaction, dynamic content, collaboration, contribution, community and social computing. These concepts have led to the development and evolution of web-based communities and hosted services such as mashups, blogs, wikis, RSS, tagging, social bookmarking and social networking (O’Reilly, 2005). An overview of the evolution of the web and its enabling technologies is shown in Figure 1.

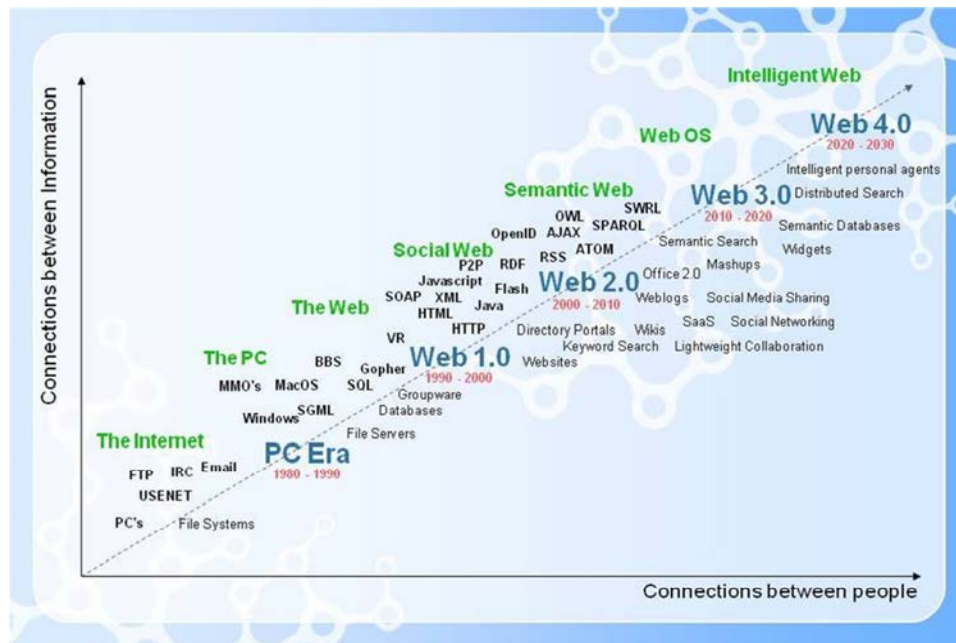
A mashup is a web-based application that is created by combining and processing online third party resources, that contributes with data, presentation or functionality (Koschmider et al., 2009). Even though Web 2.0 is often characterised by the aspect of community (social character) and user opinion expression (wikis, blogs), mashups have some important aspects that make them valuable for the Web 2.0 (Fujii, 2010; Koschmider et al., 2009).

In our application, Web 2.0 is represented by the notion of a mashup that uses web APIs, as the most common technique for mashups’ creation. Web APIs define a generic set of methods and functionalities which enable applications to call remote procedures and to exchange data by passing well-defined messages from a web service in a

transparent manner. For our application, we have chosen Amazon Web API and EBay API, since Amazon and Half.com are among the 20 top book sites (<http://books.nettop20.com>).

Towards the concept of Web 3.0, problems may emerge when there are alternative techniques by the competing visions of Web 2.0 and Semantic Web, which can serve the same purpose. Take for example linked data and web APIs, which both can be used as primary data sources for mashup development.

Figure 1 Web evolution (see online version for colours)



Source: <http://mediamaipasque.blogspot.gr/2009/10/evolution-des-technologies-du-web.html>

Besides technical details, there is a conceptual difference between web APIs and linked data. The available data items via web APIs are not assigned with globally unique identifiers. Therefore, it is not possible to set links between items in different data sources. On the other hand, applications based on linked data can draw on an unbounded, global data space. Unlike Web 2.0 mashups that can use a part or total of a fixed set of data sources, linked data mashups can discover new data sources at runtime by following data-level links, and can thus deliver more complete answers as new data sources appear on the web (Heath et al., 2009).

Nevertheless, in our work, data integration is based on the use of web APIs. The main reason is that Amazon and Half eBay resources, where our data are drawn from, are only accessible via web APIs. In addition, the methods to aggregate data by using APIs are generally considered more straightforward and have grown more technically mature, as compared to data linking (Fujii, 2010).

Even though our implementation may not consume linked data, data ingested from the APIs are triplified in web ontologies, where the application's logic is based upon. As

a step ahead, we actually make these data readily available to the LOD cloud by exposing them also as linked data, thus evidently supporting third-party applications to ‘plug-in’ and meaningfully reuse this information.

2.3 Personalisation based on semantic technologies

Generally, existing approaches to personalisation are based on three different axes (Tran et al., 2008):

- *Adaptivity dimensions*: The adaptive behaviour is realised by either collaborative-filtering (identifies content found relevant by similar users) or content-based filtering (exploits similar content to identify relevant resources for a certain user) or combination of these two.
- *Representation formalism*: This is how the aspects of the adaptive system, such as the content, the users, the system itself, are represented within a specific formalism.
- *Exploitation techniques*: Techniques which are used to perform the underlying logic of adaptation.

Many studies have pointed out the advantage of the use of semantic technologies in the aforementioned axes of personalisation approaches. The use of ontologies, in order to represent the different adaptivity dimensions, increases meaningfully the interoperability and the reusability of model information.

Especially for the user model, many efforts have been made to standardise the information about the user in terms of an ontology, such as FOAF (<http://xmlns.com/foaf/spec/>), LOM (<http://ltsc.ieee.org/wg12/20020612-Final-LOM-Draft.html>), etc. Therefore, ontologies can play many potential roles to support user modelling by:

- 1 providing a mechanism for reasoning about the users
- 2 supporting scrutability for aiding the user in getting a better understanding of the domain
- 3 defining a set vocabulary to enable metadata annotation of the content (Kay and Lum, 2005).

In the axis of exploitation techniques, rules are employed to represent the adaptation logic. Similar to ontologies, which substitute vectors, matrices, Bayesian networks, etc. as formalisms for the representation of different aspects, the usage of rules, as the logic underlying the adaptation, replaces adopted techniques from statistics and machine learning, for the same purpose. Adaptation by using rules is accomplished in a more transparent manner to the users which can better inspect and understand the entire process.

Besides the aforementioned contributions, semantic technologies also help to solve the challenging problem of developing *open-corpus* AH systems (Antoniou et al., 2005a). In a traditional closed-corpus AHS, all the documents and the relationships between them are known at the design time. However, this no longer holds when considering an open corpus of documents, and mashups can be considered to fall into this category. In this

case, Semantic Web can offer a certain *context-of-use* by allowing programmes to reason about content and its meaning (Brusilovsky and Henze, 2007).

In terms of the above, this work introduces a content-based filtering adaptation system; the representation formalism adopted is offered by web ontologies and OWL semantics; and the exploitation logic is captured by a rule-based model.

3 Related work

Our work focuses on two distinct axes:

- 1 the integration of Semantic Web and Web 2.0 so as to develop a prototype Web 3.0 application
- 2 the usage of ontologies and rules, pillars of semantic technologies, in order to achieve personalisation.

The concept of combining Semantic Web technologies and Web 2.0 has been investigated from various different angles. For example, semantic blogging constitutes an effort to enhance blogs with semantic, machine-understandable metadata and has attracted quite a lot of interest. We report indicatively some scenarios of semantic blogging such as the semiBlog editor proposed in Moller et al. (2005), the Semblog platform in Ohmukai and Takeda (2004), the prototype semantic blogging system OntoBlog (Shakya et al., 2008) and so on.

Towards the integration of mashups with the notion of the Semantic Web (Erb et al., 2009), many attempts have also been made, such as:

- 1 semantic mashup for tourism proposed by Wang et al. (2008)
- 2 semantic map mashups
- 3 semantic mashups for several scenarios in life sciences (Belleau et al., 2008; Moller et al., 2005)
- 4 the use of a mashup architecture in more sophisticated tasks, like business processes (Anjomshoaa et al., 2009).

Traditional personalisation methods, as discussed in Section 2.3, are already widely used. Semantic personalisation on the other hand has recently come to attract research focus (Vuljani et al., 2010). Similar to our approach, most of these techniques focus on the ontology-based establishment and enrichment of user profiles either on FOAF (Ankolekar and Vrandecic, 2006) or on custom methodologies (Razmerita, 2008; Tziviskou and Brambilla, 2007) and then employ content-based filtering. In Liang et al. (2008), this approach is further extended by utilising user ratings for items; however issues like the new user problem or impeding document ratings remain unresolved.

In the following, we single out and discuss a couple of semantic personalisation approaches that are most closely related to our work.

Blanco-Fernández et al. (2009) present a procedure to automatically compose interactive applications that provide personalised commercial functionalities to the users, gathering content from multiple sources and with a back-end of Semantic Web services. The procedure is driven by SWRL rules and similarity metrics based on semantic reasoning.

In addition, Wang et al. (2010) propose a tourism system based on an ontology. This system allows integration of heterogeneous online travel information and recommends tourist attractions to a user based on the Bayesian network.

The above efforts differ in two distinct dimensions, namely the representation formalism and the exploitation technique used, as discussed in Section 2. The tourist information and the content of advertised items in Wang et al. (2010) and in Blanco-Fernández et al. (2009) respectively, are both represented as basic ontology components. In Wang et al. (2010), besides the travel ontology, a user ontology is also constructed. In contrast, the user profile features in Blanco-Fernández et al. (2009) are captured in a data structure. Conversely, Wang et al. (2010) employ a Bayesian network to perform recommendation, while in Blanco-Fernández et al. (2009) Semantic Web rules decide the suitable sources for a given user.

In our work, an ontology is being used as the representation formalism both for the content sources and for user profiles. Furthermore, personalisation tasks are carried out solely by the inference power of rules. Thus, all beneficial characteristics of Semantic Web personalisation can be combined into a single unified approach. To our knowledge, there is no other intelligent mashup that performs user modelling and adaptation relying purely on the Semantic Web stack.

Finally, it should be mentioned that there is also another book mashup (Bizer et al., 2007) which integrates book data from web APIs into the Semantic Web. However, it appears to lack personalisation features.

4 System architecture and design

In this section, we describe the 3-tier architecture, which constitutes the base of our application and may also underlie many sophisticated Web 3.0 applications (Pomonis et al., 2009). Then, we focus on the design of the ontology and the philosophy behind our personalisation rules.

4.1 Architecture overview

As illustrated in Figure 2, the layers of the architecture can be distributed both at the logical and physical level:

- 1 the front-end layer
- 2 the application logic layer
- 3 the knowledge management layer.

The knowledge base of our application resides on the lower part of the architecture, namely the knowledge management layer. This is represented by our core ontology, *BookShop*, which includes rules for personalisation and the individuals (instances of classes).

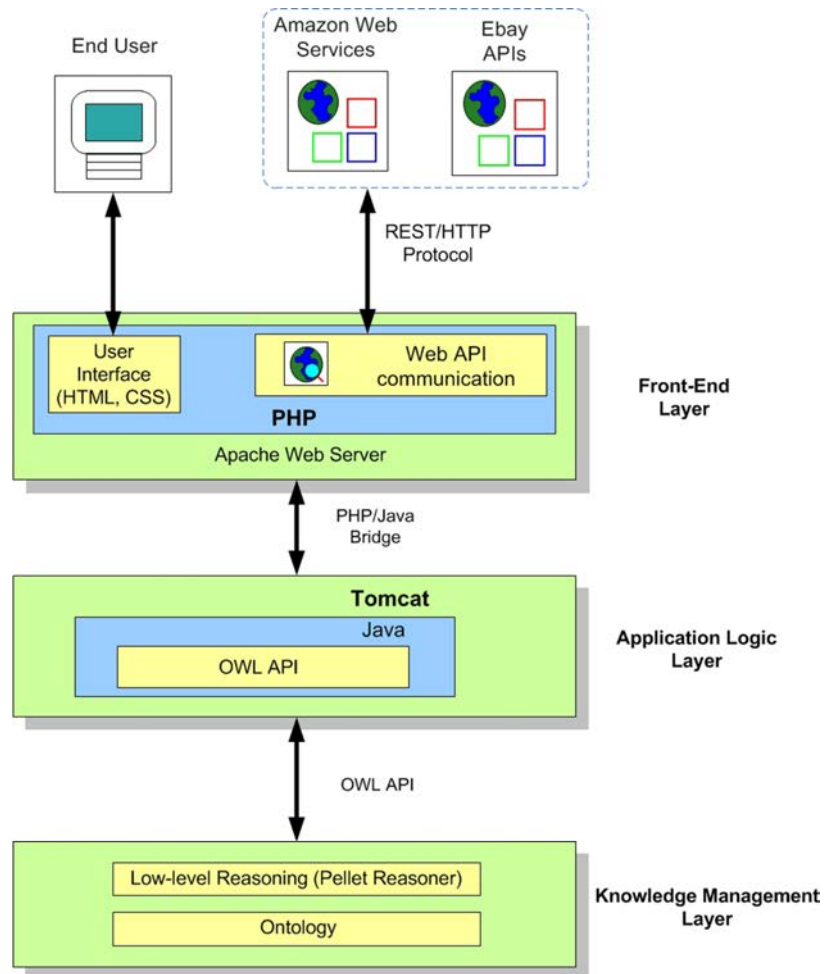
Then, the middle layer of the 3-tier architecture, the *application logic layer*, is responsible for managing and uploading the ontological model of the lower layer. The

operation of this layer is implemented with the use of the OWL API (Horridge and Bechhofer, 2009) that serves also as a means of communication with the lower, knowledge management layer. The middle layer is also responsible for the ontological data loading, coming through the front end and based on the ontological schema of the back end.

Finally, the upper layer of the proposed architecture is in general the layer which enables users to fully interact with the knowledge base, by adding, eliciting and incrementing the ontological data model. This layer may also interact with web services, programmes, scripts, and other interoperability interfaces. It performs the following tasks:

- communication/interaction of the application with the Amazon and Half eBay web APIs
- user interface of the application with the ability of interaction with the knowledge base, presentation and navigation in semantic data.

Figure 2 The 3-tier architecture (see online version for colours)



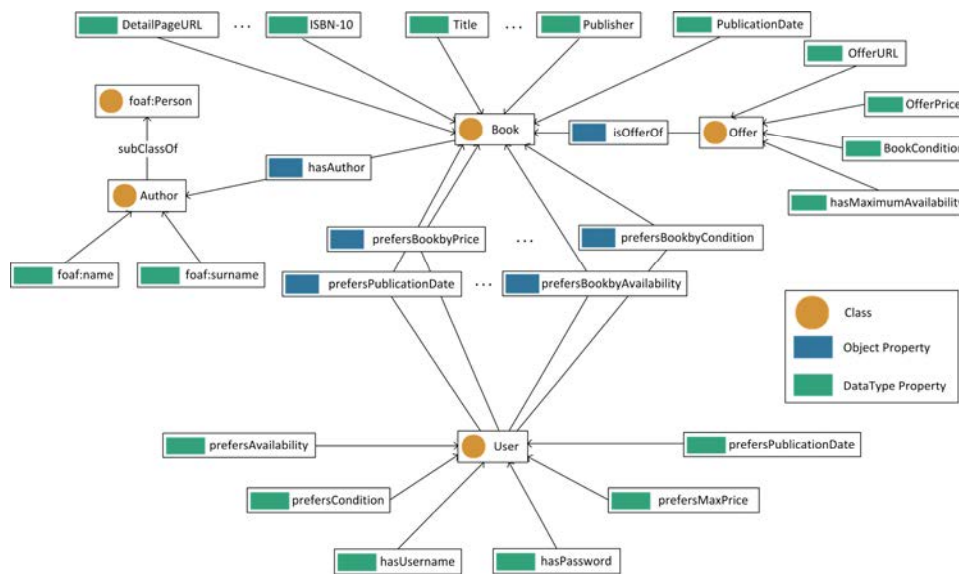
4.2 BookShop ontology

To design our book ontology, we first took into account the kind of metadata offered by Amazon and Half eBay responses. Our design process has resulted in the core ontology BookShop and part of this is shown in Figure 3. BookShop contains four main classes: *Book*, *Author*, *Offer* and *User*.

The class *Book* is enriched with relations such as title, publisher, dimensions, ISBN, publication year, number of pages, format, sales rank, images in various sizes and a URL – corresponding to the Amazon online bookstore – so as to describe the instances of this class. All these relations are captured in the ontology as datatype properties.

For our purposes, keeping record of each author's name and surname suffices. Therefore, we decided to define the class *Author* as a subclass of the class *Person*, which is included in the friend of a friend (FOAF) ontology. FOAF provides a unified way to describe persons, expressing their interests, their activities and their relations to other people and objects. The FOAF properties *foaf:surname* and *foaf:firstName* have been used to model author last and first name.

Figure 3 BookShop ontology (see online version for colours)



Items for sale on Amazon and Half eBay can be sold by more than one seller for different prices and in different conditions ('new' or 'used'). Thus, any item – any book in this case – is associated with one or more offers. An offer is a combination of price, availability, condition and vendor/seller. Therefore, to find a book's price, we have to get the offers made by the vendors selling the book on online bookstores. The concept of an offer is represented by the class *Offer*. Datatype properties such as *BookCondition*, *hasMaximumAvailability*, *OfferPrice* and *WebSiteOrigin* express the condition, the availability, the price of the offered book and the seller URL in these bookstores, respectively.

The class *User* is meant to express user profiles. We capture the preferences of each user in this class, such as preferable condition, preferable minimum availability,

preferable minimum publication year and preferable maximum price (preference criteria). All this data about users are represented as datatype properties. Additional preferences can be easily added by simply introducing additional datatype(or object) properties, exactly like the ones mentioned above. Possible preferences that can be added depend on the relations of the *Book* class and include characteristics such as book format, language and category.

The *User* class and related properties in the ontology reflect a snapshot of the preferences for each register user. Changes in user behaviour are easily captured by allowing the users to interact with the ontology, through the ‘settings’ menu option that effectively allows the update or modification of their preferences.

Relationships between instances of the BookShop ontology are represented by object properties that express relations between instances of two classes. In this context, a book must have at least one author (hasAuthor and inversely isAuthorOf) and there is at least one offer for a book (isOfferOf and inversely hasOffer).

Object properties that link an instance of the class *Book* to an instance of the class *User* and inversely, are defined to express the user’s preference for a book depending on which preference fields of the user’s profile are covered. Then, a set of DL-safe SWRL rules is responsible for actually populating these properties. For example, the object property prefersBookbyCondition would relate a user with books being in a condition the user prefers. There are also properties for the case when more than one preference criteria are met. For example, prefersBook_byRate2 would hold in case two criteria are satisfied together (no matter which). Instead of implementing this within programme code, we chose to rely solely on the expressive power of ontology properties and corresponding rules.

4.3 Rule-based personalisation

In order to obtain personalisation, a set of DL-safe rules was written, using SWRL. These rules ‘match’ user’s preferences (user profile) with the features of books, which are returned after searching Amazon and Half eBay web services. These ‘personalisation rules’ essentially distinguish those books, which satisfy user’s preferences from the entire set of books after the searching process. Naturally, each user preference has a direct, one-to-one relationship to a personalisation rule. Therefore, in accordance to the preference criteria introduced in 4.2, four rules were written to check the satisfiability of each preference criterion separately. Additional personalisation criteria can be easily accommodated by adding a new personalisation rule.

Take for example the case of a rule about preferred book condition. The SWRL description for this rule is given by Rule #1 in Table 1. Similarly, a SWRL rule was defined in order to associate a user with books having the user’s preferred max price (Rule #2). The rule about a user, which prefers books with time availability equal or smaller than a certain interval, is described by Rule #3. Finally, the user preference for a book published during or after a certain publication year can be illustrated by Rule #4.

For the case where two or three or four preference criteria are satisfied together, we wrote more rules so as to check the number of satisfied criteria (Rule #5). There are also more rules to check all possible combinations.

As an example, Rule #5 can be translated in natural language as follows: “If a user prefers books priced no more than 40.00 USD, published in 2000 or newer, and delivered

in less than 50 hours, then a book matching all these conditions would be linked to this user via the object property `prefersBook_byRate3`".

Table 1 SWRL rules examples

Rule #1	Book(?y) AND Offer(?z) AND User(?x) AND isOfferOf(?z,?y) AND BookCondition(?z,?condition) AND prefersCondition(?x,?preferred_condition) AND swrl:equal(?condition,?preferred_condition) → prefersBookbyCondition(?x,?y)
Rule #2	Book(?y) AND Offer(?z) AND User(?x) AND isOfferOf(?z,?y) AND OfferPrice(?z,?price) AND prefersMaxPrice(?x,?max_price) AND swrlb:lessThanOrEqual(?price,?max_price) → prefersBookbyPrice(?x,?y)
Rule #3	Book(?y) AND Offer(?z) AND User(?x) AND isOfferOf(?z,?y) AND hasMaximumAvailability(?z,?max_availability) AND prefersAvailability(?x,?prefer_availability) AND swrlb:lessThanOrEqual(?max_availability,?prefer_availability) → prefersBookbyAvailability(?x,?y)
Rule #4	Book(?y) AND User(?x) AND PublicationDate(?y,?date) AND prefersPublicationDate(?x,?preferred_date) AND swrlb:lessThanOrEqual(?preferred_date,?date) → prefersBookbyPublicationDate(?x,?y)
Rule #5	Book(?y) AND User(?x) AND prefersBookbyPrice(?x,?y) AND prefersBookbyPublicationDate(?x,?y) AND prefersBookbyAvailability(?x,?y) → prefersBook_byRate3(?x,?y)

5 Collecting data from bookstores

In this section, we review the process of searching data about books from the web data sources, in other words the application's interaction with Amazon and Half eBay web APIs. A diagram that depicts data flow upon interaction with the Web APIs, appears in Figure 4. Whenever the user sends a *searching call*, the searching process starts to query data from Amazon Web Services (AWS), and especially from the *US E-Commerce Service (ECS)*. In order to extract the appropriate data for our application, we choose the *ItemSearch* operation, among the set of available ECS operations. The user search keywords are passed as parameters to this operation to ensure the relevance of results to the typed keywords. Additional parameters are also set, in order to limit the result-set to books only.

A request to Amazon may return many thousands of items in a response. Returning all these results at once may be inefficient and impractical. In order to alleviate this, we make successive calls and combine responses into a single XML file. Results are inserted into this XML file in the order in which they are received. Therefore, results are fetched within our application already ranked and ordered by virtue of Amazon's respective algorithms.

The consolidated XML file is then processed by an XSLT in order to rule out redundant data that are not useful in our implementation. For example, information such as images height and width, number of used, new, collectible or refurbished books, has no added value for our application.

Once our application completes the search process at Amazon, it starts searching Half eBay: for each book returned by Amazon, we find additional offers that may be available at Half eBay. We use the *eBay shopping Web Services* and particularly, the *FindHalfProducts* operation. The interaction with the eBay shopping API is based also on the REST-protocol and the exchange of URL requests and XML files-responses.

Figure 4 Diagram of communication with web APIs (see online version for colours)

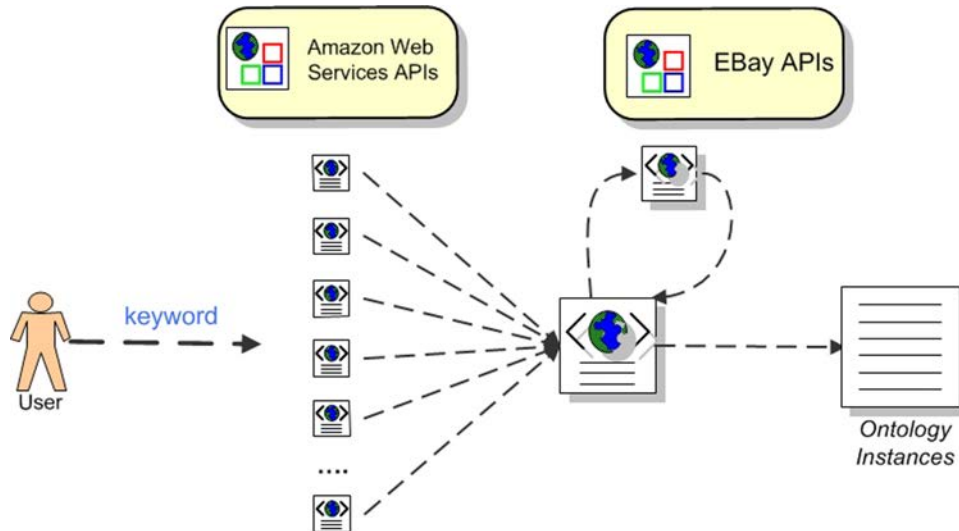


Figure 5 Sample record of the XML file

```
<Book>
  <DetailPageURL>
    http://www.amazon.com/Water...
  </DetailPageURL>
  <Author>Sara Gruen</Author>
  <ISBN>1565125606</ISBN>
  <NumberOfPages>350</NumberOfPages>
  <PublicationDate>2007-04-09</PublicationDate>
  <Publisher>Algonquin Books</Publisher>
  <Title>Water for Elephants: A Novel</Title>
  <SalesRank>100</SalesRank>
  <Offer>
    <GlancePage>
      http://www.amazon.com/gp/help/seller/h...
    </GlancePage>
    <Condition>Used</Condition>
    <Price>
      <Amount>698</Amount>
      <CurrencyCode>USD</CurrencyCode>
    </Price>
    <FormattedPrice>$6.98</FormattedPrice>
  </Offer>
  ...
</Book>
```

Figure 6 Individual of class author

```

<Author rdf:ID="Author_id2134438">
  <foaf:firstNamerdfs:datatype="http://www.w3.org/2001/XMLSchema#Literal">
    Sara
  </foaf:surnamerdfs:datatype="http://www.w3.org/2001/XMLSchema#Literal">
    Gruen
  </foaf:surname>
  <isAuthorOfrdf:resource="#Book1"/>
</Author>

```

When the searching process at the Half eBay is complete, results are once more unified into a single XML file (Figure 5). In order to do this, we actually append more offer sub-elements for each book element into the Amazon's results document. Again, Half eBay offers are inserted into the XML file following their original ordering. Finally, the elements of this XML file are converted into an OWL file with individuals and matched against our BookShop ontology schema. An excerpt of the final OWL file is shown in Figure 6.

6 Functionality and usage

In this section, we demonstrate the functionality of the application. Moreover, we outline an indicative usage case in order to point out its capabilities and features.

6.1 Functionality

The first page of the application includes two choices for the visitor, 'new user' and 'registered user'. New users are presented with a completion form, which includes fields such as *book condition*, *maximum book price*, *publication year* and *maximum book availability*, thus forming the user profile.

After successful authorisation, the user is directed to the main page of the application, which includes a search form. For efficiency and demonstration purposes there is also the choice to select between 'search on the fly' and 'prefetched search', where the user interacts with a prefetched dataset. The search form consists only of a 'Search' button and a text field, where the user types in some keywords to initiate the book searching process (*searching call*, Section 5).

When searching ends, results are imported into the core BookShop ontology as individuals. As per the data collection procedure, the ranking of results at this point follows and depends solely on Amazon's and Half eBay's respective algorithms. Next, the ontology is classified by the reasoner and the rules, that would determine how user preferences are matched, are fired against the ontology. Search results are effectively re-ranked based on the rules outcome, as shown in Figure 7. In particular, the more criteria a book satisfies (the more preference rules it triggers), the higher it appears in the results. In essence, results are grouped into four distinct sets depending on the number of preference criteria matched. The group members' relevance to the search keywords is guaranteed by the original data providers.

Figure 7 Presentation of preferred books for User_1 (see online version for colours)

Welcome User_1

Search Settings Logout

Favourite Books

ID	Book's title	Score
1	iPad: Visual QuickStart Guide (Visual Quickstart Guides)	!!!!
2	iPad For Dummies (For Dummies (Computers))	!!!!
3	The Hunger Games: Movie Tie-in Edition	!!!!
4	My iPad for Kids (Covers iOS 6 on iPad 3rd or 4th generation, and iPad mini) (2nd Edition)	!!!!
5	Your iPad at Work (Covers iOS 6 on iPad 2, iPad 3rd/4th generation, and iPad mini) (3rd Edition)	!!!!
6	iPad For Seniors For Dummies (For Dummies (Computer/Tech))	!!!!
7	iPad for Seniors: Get Started Quickly with the User Friendly iPad (Computer Books for Seniors series)	!!!!
8	iPad: The Missing Manual	!!!!
9	A Newbies Guide to Using GarageBand For the iPad	!!!
10	A Newbies Guide to Using iMovie For the iPad	!!!

Previous Page Page 1 to 10 Next Page

Each table's row includes information, such as an auto-incrementing number, the book's title and a number of exclamation marks which express the number of satisfied criteria. Each book's title is a link and clicking it, a page appears with all the available features of the specific book, like title, author, ISBN, etc. (Figure 8). The title and image of the book are links to its 'official' page at Amazon and the available offers for this book are presented in a table. Following the *search* link (see Figure 7), the user is taken back to the main page in order to initiate a new query.

Figure 8 Book information (see online version for colours)

 <p>iPad For Dummies (For Dummies (Computers)) Sales Rank in Amazon Books: <i>Not available</i> Book's Details - Author : Edward C. Baig , Bob LeVitus - ISBN : 0470580275 - Publisher : For Dummies - Publication Year : 2010 - Format : Paperback - Number of Pages : 336 - Transport Weight : 170 Pounds - Book's Dimensions : 90 X 910 X 730</p> <p>RDF</p>	ID	Condition	Price	Web Origin	Availability in hours
	1	New	5.99	amazon.com	Not available
	2	New	56.95	ebay	Not available
	3	Old	2.99	ebay	Not available
	4	Old	3.0	ebay	Not available
	5	Old	3.18	ebay	Not available

Finally, book information is made available as linked data in RDF, where books' and offers' links are resolvable within our application or within their original context, respectively (Figure 9). This is also true for the search results list, which is exposed as RDF along with the inferred book rankings, for other applications to be able to consume and reuse.

Figure 9 Linked data (see online version for colours)



6.2 An example usage case

This subsection contains a brief example of Books@HPCLab personalisation features, so as to demonstrate that different users can be treated differently, even though they are searching for the same key-phrase. We shall consider two users, Mary and George. According to Mary's profile, she prefers books that were published no earlier than 2009, their price does not exceed 15.00 USD, their condition is 'new' and they can be delivered within 50 hours. On the other hand, George's most favourite books should be in 'old' condition, their price should be lower than 45.50 USD, have been published at least in 2000 and they can be delivered within 12 hours.

Mary and George are searching for books with the topic *Semantics*, so they are typing in the same keyword: 'semantics'. At this moment, the application gets a lookup call, so it decodes the keyword and starts to query the original data sources (Amazon and Half eBuy). When the collection data process is complete, a XML file with 100 book nodes is returned in both cases, with the same content, as it is expected.

In the case of George, the procedure goes on as follows:

- Our DL reasoner, Pellet, finds that, for the books returned by the query, George's profile matches the conditions of those SWRL rules which apply when two preference criteria are satisfied together (Subsection 4.3).
- Next, Pellet finds the applicable SWRL rules for George, which satisfy just one of the four preference criteria. None of the other SWRL rules is applicable for George.

Table 2 Application response for George

<i>Number of returned books</i>		74
Set 1	Number of returned books which satisfies exactly two criteria	2
Set 2	Number of returned books which satisfies exactly one criterion	72

The total number of books, which are shown to George is 74, therefore there are 26 books returned from the collection process, which satisfy none of George's preference criteria. From the 74 books, there are only two which satisfy exactly two criteria and the rest of them satisfy just one criterion at a time (see Table 2).

On the other hand, the procedure for Mary goes on as follows:

- The reasoner finds that Mary's profile matches the conditions of those SWRL rules, which check that three preference criteria are satisfied together (Subsection 4.3). So the instance Mary is being related to six instances of the class Book through the property *prefersBook_byRate3*.
- In addition, the rules, which check the simultaneous satisfiability of two preference criteria, are triggered.
- Next, Pellet finds the applicable SWRL rules for Mary, which satisfy just one of the four preference criteria. For Mary, there is no book that satisfies all four preference criteria at the same time.

Table 3 summarises the total number of returned books and the number of books with the respective amount of satisfied criteria in the case of Mary.

Table 3 Application response for Mary

<i>Number of returned books</i>		99
Set 1	Number of returned books which satisfies exactly three criteria	6
Set 2	Number of returned books which satisfies exactly two criteria	71
Set 3	Number of returned books which satisfies exactly one criterion	22

Consequently, our application returns more books to Mary and more of Mary's preference criteria are satisfied together, in contrast to George. Finally, there are books, which are returned both to Mary and George, but in different order. Take for example the book with title 'HTML Mastery: semantics, standards and styling', which belongs to Set 2 (Table 3) in Mary's case. In the case of George, this book appears in the set of books that satisfy only one preference criterion (Set 2 in Table 2).

7 Evaluation

The evaluation of semantic systems services is still a problematic process and known IR evaluation patterns (like recall and precision) are often not well-suited or insufficient for this purpose (for example see, Strasunskas and Tomassen, 2010). Therefore, in order to identify the major points made by the design of the Books@HPCLab semantic mashup application, we measure the performance of this mashup and evaluate it based on three axes, mainly reflecting a user-centred perspective of the system: *personalised search accuracy*, *response time* and *ontology metrics*. In the following, we present in detail the

evaluation procedure, which comprises of a set of experiments within our specific evaluation framework, as well as the methodology of the experiments and their configuration. Finally, we discuss the results for each of the three evaluation dimensions.

7.1 Procedure

In order to outline the context of the evaluation procedure, we begin with the description of participants and their interaction with the system. A sample of 10 users participated in the experiments and specified manually their profiles. Each participant is required to issue 10 test keyphrases in the search field of the application so as to retrieve book information from Amazon and Half eBay, which is relevant to these keywords. We remind that relevance of results at this stage is determined solely by the data providers' algorithms. After the end of the search task, participants browse through search results and click on those books that presumably contain information that match their profile. Naturally, books that are skipped are those in which the participant is not interested. With this approach, we ensure that the relevance of the search results is specified each time by the participants.

The set of test keywords is not selected in a random manner. We tried to find out the list of the most common keywords that users of Amazon type when searching for books. To do this, we utilised Amazon's search box autocomplete feature, assuming that the most frequently keyphrases are suggested. Table 4 shows the queries contained in the test set:

Table 4 Most frequently search keywords in Amazon.

Q1	diary of a wimpy kid	Q6	obama
Q2	gone girl	Q7	vince flynn
Q3	harry potter	Q8	x-men
Q4	ipad	Q9	yoga
Q5	james patterson	Q10	zombie

7.2 Metrics

To estimate *personalised search accuracy*, a modified version of the average rank metric (Qiu and Cho, 2006; Speretta and Gauch, 2005) is used for its measurement. The average rank of a keyword s taking into account all users' actions is defined as below:

$$AvgRank_s = \frac{\sum_{p \in P_{s_1}} R(p) + \sum_{p \in P_{s_2}} R(p) + \dots + \sum_{p \in P_{s_{10}}} R(p)}{P_{s_1} + P_{s_2} + \dots + P_{s_{10}}}$$

Here, P_{s_i} denotes the number of books that the user i has clicked for the test keyword s and $R(p)$ denotes the rank of book p . In this version, $R(p)$ can take any of the following distinct values in $C = \{1, 2, 3, 4\}$, since books are ranked depending on the number of user preference criteria that are satisfied.¹ For example, if user #1 has clicked the book p that satisfies only three preference criteria, then the value of $R(p)$ is 3. The final average rank on test keyword set S is computed as:

$$AvgRank = \frac{1}{|S|} \sum_{s \in S} AvgRank_s$$

In our case, the value of average rank indicates better placements of relevant results or better result quality, when it is closest to 4. We can also normalise this value so as not to depend on the number of criteria:

$$AvgRank_{norm} = \frac{AvgRank}{|C|}$$

A value of 1 ideally means that all search results clicked by the users are the ones with the highest ranking in our mashup. The AvgRank computed for our system using this method is then compared to the AvgRank of well-known search systems, e.g., Bing or Google. To make them comparable, we simply consider their inverse, i.e., the reciprocal rank.

Another major point we used to evaluate our application is *response time*. In order not to account for any data providers' latencies and to better bind the workload of our implementation, we conducted our experiments against prefetched datasets for the particular queries involved. Therefore, in the context of our application, response time is considered as the total of two components, *query time* and *reasoning time*. With query time, we mean the time it takes for our system to query the inferred ontology model and render search results to the user, while reasoning time is the time it takes for the reasoner to classify the ontology and evaluate the personalisation rules against the dataset.

Finally, to give an estimate of the relative size of the problem-set handled by the application we also report a series of *ontology metrics*. More precisely, we write down the number of classes, object/datatype properties and rules found in the BookShop ontology, as well as the number of individuals and assertions which the ontology is populated with, after data fetching for a single query is complete. The reasoner then processes the rules and infers additional assertions, thus expanding the knowledge base. The size of this additional knowledge is also reported, in order to give an approximation of rules productivity and overall efficiency of the reasoning process.

7.3 Results

The following figure reports the average rank for each of the 10 test queries, as well as the overall and normalised AvgRank. We notice an $AvgRank_{norm}$ of 0.922, which means that, on average, relevant results are within the highest (4 criteria) or second-to-highest (3 criteria) ranking groups. In turn, this suggests that the ontology-based profiles and rules are able to deliver personalised search results that highly stand up to user preferences. To have an appreciation of the meaning of such ranking scores, independent studies for Bing and Google report a reciprocal rank of 0.898 and 0.932, respectively (Liu, 2012), or even lower (Kumar et al., 2011; Dou et al., 2007).

Next, Table 5 summarises response times, Table 6 reports ontology metrics for the BookShop ontology schema and Table 7 includes the average individual and assertion count for a sample query over all participants, right before and after reasoning.

Query times are submultiple of reasoning times, with the latter amounting to a few seconds. This happens because the reasoning process is a hard computational task, while querying the knowledge base is relatively straightforward, since all results are directly

available and all possible inferences have been made. Nevertheless, we notice that the reasoning process contributes an almost 33% of new assertions, with the object property assertions having almost tripled, meaning that our particular rule-set is highly productive and that it sufficiently accommodates the implications of user preferences.

Table 5 Response times

	<i>Q1</i>	<i>Q2</i>	<i>Q3</i>	<i>Q4</i>	<i>Q5</i>	
Reasoning time (ms)	10,182	17,175	15,377	15,047	17,448	
Query time (ms)	291	687	546	489	707	
	<i>Q6</i>	<i>Q7</i>	<i>Q8</i>	<i>Q9</i>	<i>Q10</i>	<i>Mean</i>
Reasoning time (ms)	13,439	10,919	15,628	15,244	8,990	13,945
Query time (ms)	421	320	603	492	252	481

Figure 10 Average normalised rank for the experiment queries (see online version for colours)

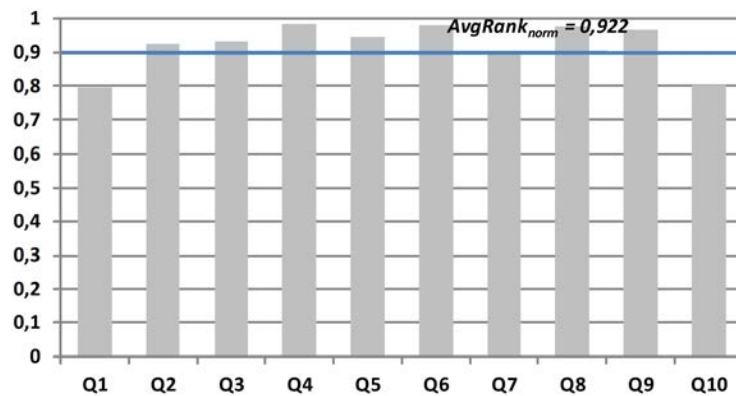


Table 6 Metrics for the ontology schema

<i>Ontology schema</i>	
Classes	33
Object properties	81
Datatype properties	63
Rules	33

Table 7 Average counts before and after reasoning

	<i>Populated ontology</i>	<i>Inferred ontology</i>
Object property assertions	2,646	6,543
Datatype property assertions	9,331	9,340
Individuals	1,425	1,425

8 Discussion and future directions

In this work, we have described a concrete scenario of how Semantic Web technologies could enhance Web 2.0 tools and especially mashups. The resulting semantic mashup has been augmented with personalisation features based on the use of rule filtering as a recommendation technique.

The integration of semantics and, mainly, ontologies within Web 2.0 applications has also been studied before (Section 3). Light-weight ontology languages like RDF are often easier to handle, both for machines and for humans, than more complex formalisms like OWL. The fact that even simple machine-readable data can bring benefits, may lead someone to believe that the Semantic Web does not need more expressivity than RDF. However, depending on practical uses, further expressivity is necessary to express complex knowledge. Examples of this set of applications include semantic mashups in life sciences or in business areas.

Our engineering effort to enhance our book mashup application with web semantics is primarily based on the development of an OWL ontology. In contrast to Bizer et al. (2007), which uses RDF, we use much more complex formalisms to extend our mashup ontology with rules so as to achieve personalisation. We have shown that the high rate of knowledge added is worth this effort and this at a relatively low time cost.

The usage of rules, as an adaptation logic technique, achieves a substantial level of personalisation in a manner more understandable and transparent to users. This is also confirmed by the increased average rank scores in our experiment, as opposed to, i.e., traditional search engines. The power of ontologies in such personalised mashups lies in the interoperability and in the reusability of information as well as in a sound and expressive logic framework to perform fine-grained adaptations upon.

As a lesson-learned, we advise that it is not desirable to specify a too large number of rules, in an attempt to check all the possible combinations of user preference criteria. Although this can be feasible and effective, most of the time just a few rules are necessary and the rest of the task can be accomplished programmatically.

In terms of performance, the manipulation and merging of large XML files using DOM can often be slow. A relational database or other persistent store for keeping XML responses can be used as an intermediary data source for populating ontology documents dynamically, as well as for caching purposes. Alternative exchange formats can also be investigated, such as JSON or the asynchronous processing of XML documents.

Finally, Books@HPCLab treats all rules with equal importance. It might be worth to consider putting weights on rules in a fashion similar to Pan et al. (2005), though not ranking rules in terms of certainty, but in regard to their individual value in a particular user's profile. These weights could be user-defined and depend on user scores or implicitly adjusted, as a means to produce adaptations based on observable user behaviour.

9 Conclusions

In this work, we have shown that Semantic Web and Web 2.0 can be complementary visions for the future of the web, rather than in competition. This was achieved by the development of an application which unifies successfully the philosophy of Web 2.0 applications (mashup) and the powerful technical infrastructure of the Semantic Web

(ontologies and rules). Such web applications are considered to be part of the next generation web, usually referred to as Web 3.0.

In particular, we presented a prototype web application, which integrates information from web APIs, such as Amazon Web API and Half eBay API, converting them to individuals of an ontology schema, concluding in a kind of semantic mashup for books. At the same time, information are gathered from their original sources, acquire semantic structure and exposed as linked data. The integration of rules into this semantic mashup serves as a viable and straightforward alternative towards personalisation features.

Our literature survey reveals that a strong interest exists in filtering content ‘mashed up’ from various sources using semantics; this is mainly due to a growing need for eliciting value and meaning out of the plethora of unauthoritative and community-oriented information on today’s Web 2.0. To this end, the Semantic Web and the LOD paradigm appear to be worth investigating as almost natural means to achieve these goals. We believe that our dealing with the various conceptual and technological challenges and the lessons-learned in this process may serve as a useful guidance towards developing tailored Semantic Web applications in the Web 3.0 framework.

References

- Anjomshoaa, A., Bader, G. and Tjoa, A. (2009) ‘Exploiting mashup architecture in business use cases’, *IEEE International Conference on Network-Based Information Systems*, Indianapolis, USA (invited).
- Ankolekar, A. and Vrandecic, D. (2006) ‘Personalizing web surfing with semantically enriched personal profiles’, in Bouzid, M. and Henze, N. (Ed.): *Proc. Semantic Web Personalization Workshop*, Budva, Montenegro.
- Antoniou, G., Baldoni, M., Baroglio, C., Baumgartner, R., Bry, F., Henze, N., May, W., Patti, V. and Wierzhon, S. (2005a) ‘Personalization for the Semantic Web II’, Technical Report IST506779/Hannover/A3-D4/D/PU/b1, Reasoning on the Web with Rules and Semantics, REWERSE.
- Antoniou, G., Damásio, C., Grosz, B., Horrocks, I., Kifer, M., Maluszynski, J. and Patel-Schneider, P.F. (2005b) ‘Combining rules and ontologies’, Technical Report IST506779/Linköping/I3-D3/D/PU/a1, Linköping University.
- Bechhofer, S., Van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F. and Stein, L.A. (2004) *OWL Web Ontology Language Reference, W3C Recommendation* [online] <http://www.w3.org/TR/2004/REC-owl-ref-20040210/> (accessed 25 November 2012).
- Belleau, F., Nolin, M., Tourigny, N., Rigault, P. and Morissette, J. (2008) ‘Bio2RDF: towards a mashup to build bioinformatics knowledge systems’, in *Journal of Biomedical Informatics*, Vol. 41, No. 5, pp.706–716.
- Berners-Lee, T., Hendler, J. and Lassila, O. (2001) ‘The Semantic Web’, in *Scientific American Magazine*, Vol. 284, No. 5, pp.33–43.
- Bizer, C., Cyganiak, R. and Gauß, T. (2007) ‘The RDF book mashup: from web APIs to a web of data’, *3rd Workshop on Scripting for the Semantic Web*.
- Blanco-Fernández, Y., López-Nores, M., Pazos-Arias, J. and Martín-Vicente, M. (2009) ‘Automatic generation of mashups for personalized commerce in digital TV by semantic reasoning’, *4th International Workshop on Semantic Media Adaptation and Personalization*.
- Brusilovsky, P. and Henze, N. (2007) ‘Open corpus adaptive educational hypermedia’, in Brusilovsky, P., Kobsa, A. and Nejdl, W. (Eds.): *The Adaptive Web*, Springer.

- Cho, A. (2008) *What is Web 3.0? The Next Generation Web: Search Context for Online Information* [online] http://internet.suite101.com/article.cfm/what_is_web_30 (accessed 25 November 2012).
- Dou, Z., Song, R. and Wen, J.R. (2007) 'A large-scale evaluation and analysis of personalized search strategies', in *16th International World Wide Web Conference (WWW2007)*.
- Erb, B., Elsholz, J. and Hauck, F. (2009) *Semantic Mashup: Semantic Mashup, Mashing Up Information in the Today's World Wide Web – An Overview*. VS-R08.
- Fujii, A. (2010) 'Expanding use of web API – vast potential of mashup', in *Science and Technology Trends: Quarterly Reviews*, Science and Technology Foresight Center, NISTEP, Vol. 36, pp.10–19.
- Heath, T., Hepp, M. and Bizer, C. (2009) 'Linked data – the story so far. special issue on linked data', *International Journal on Semantic Web and Information Systems*, Vol. 5, No. 3, pp.1–22.
- Hitzler, P. and Parsia, B. (2009) 'Ontologies and rules', in Staab, S. and Studer, R. (Eds.): *Handbook on Ontologies*, 2nd ed., pp.111–132, Springer.
- Horridge, M. and Bechhofer, S. (2009) 'The OWL API: a Java API for working with OWL 2 ontologies', *6th OWL Experiences and Directions Workshop*.
- Kay, J. and Lum, A. (2005) 'Ontology-based user modelling for the Semantic Web', *10th International Conference on User Modelling, Workshop*, Vol. 8, pp.11–19.
- Koschmider, A., Torres, V. and Pelechano, V. (2009) 'Elucidating the mashup hype : definition, challenges, methodical guide and tools for mashups', in *Proc. of the 2nd Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009)*, Madrid, Spain.
- Kumar, H., Park, S. and Kang, S. (2011) 'A personalized URL re-ranking methodology using user's browsing behavior', in *5th KES International Conference, KES-AMSTA 2011*, pp.212–220.
- Lassila, O. and Handler, J. (2007) 'Embracing Web 3.0', *IEEE Internet Computing*, Vol. 11, No. 3, pp.90–93.
- Liang, T.P., Yang, Y.F., Chen, D.N. and Ku, Y.C. (2008) 'A semantic expansion approach to personalized knowledge recommendation', in *Decision Support Systems*, Vol. 45, No. 3, pp.401–412.
- Liu, Y. (2012) *Evaluation Report for COS435 'Retrieval, Discovery and Delivery' Assignment 2 in Department of Science*, University of Princeton [online] http://www.cs.princeton.edu/courses/archive/spring12/cos435/Probs/ps2_report.html (accessed 25 November 2012).
- Moller, K., Breslin, J. and Decker, S. (2005) 'semiBlog-Semantic Publishing of DesktopData', *14th Conference on Information Systems Development*.
- O'Reilly, T. (2005) *What Is Web 2.0, Design Patterns and Business Models for the Next Generation Software* [online] <http://oreilly.com/web2/archive/what-is-web-20.html> (accessed 25 November 2012).
- Ohmukai, T. and Takeda, H. (2004) 'Semblog: personal publishing platform with RSS and FOAF', *1st workshop on Friend of a Friend, Social Networking and the (Semantic) Web, Proceedings*, Galway, pp.217–221.
- Pan, J., Stamou, G., Tzouvaras, V. and Horrocks, I. (2005) 'f-SWRL: a fuzzy extension of SWRL', in *Artificial Neural Networks: Formal Models and their Applications – ICANN 2005 Lecture Notes in Computer Science*, Vol. 3697, pp.829–834.
- Pomonis, T., Koutsomitropoulos, D., Christodoulou, S. and Papatheodorou, T. (2009) 'Towards Web 3.0: a unifying architecture for next generation web applications', in Murugesan, S. (Ed.): *Handbook of Research on Web 2.0, 3.0 and X.0: Technologies, Business and Social Applications*, IGI Global, pp.192–204.
- Qiu, F. and Cho, J. (2006) 'Automatic identification of user interest for personalized search', in *Proceedings of the 15th International Conference on World Wide Web (WWW)*, pp.727–736.

- Razmerita, L. (2008) 'Modeling behavior of users in adaptive and semantic-enhanced information systems: the role of a user ontology', *Authoring of Adaptive and Adaptable Hypermedia Workshop, in Conjunction with Adaptive Hypermedia Conference*, Hanover.
- Shakya, A., Wuwongse, V., Takeda, H. and Ohmukai, I. (2008) 'OntoBlog: informal knowledge management by semantic blogging', *2nd International Conference on Software, Knowledge, Information Management and Applications, SKIMA*, pp.197–202.
- Speretta, M. and Gauch, S. (2005) 'Personalized search based on user search histories', in *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pp.622–628.
- Strasunskas, D. and Tomassen, S.L. (2010) 'On variety of semantic search systems and their evaluation methods', in *Proceedings of the International Conference on Information Management and Evaluation, Academic Conferences International*, pp.380–387.
- Tran, T., Cimiano, P. and Ankolekar, A. (2008) 'A rule-based adaptation model for ontology-based personalization', in *Advances in Semantic Media Adaptation and Personalization. Studies in Computational Intelligence*, Springerlink, Vol. 93, pp.117–135.
- Tziviskou, C. and Brambilla, M. (2007) 'Semantic personalization of web portal contents', in *WWW '07: Proceedings of the 16th International Conference on World Wide Web, USA*, pp.1245–1246.
- Vuljani, D., Rovani, L. and Baranovi, M. (2010) 'Semantically enhanced web personalization approaches and techniques', in *32nd IEEE International Conference on Information Technology Interfaces (ITA)*, pp.217–222.
- Wang, W., Zeng, G. and Tang, D. (2010) 'Bayesian intelligent semantic mashup for tourism', *Concurrency and Computation: Practice and Experience*.
- Wang, W., Zeng, G., Zhang, D., Huang, Y., Qiu, Y. and Wang, X. (2008) 'An intelligent ontology and Bayesian network based semantic mashup for tourism', in *IEEE Congress on Services*, pp.128–135.

Notes

- 1 In standard AvgRank, $R(p)$ can be any integer between 1 and the total number results, i.e., the result's position.