

# Intelligent Book Mashup : Using Semantic Web Ontologies and Rules for User Personalisation

Aikaterini K. Kalou Tzanetos Pomonis Dimitrios Koutsomitropoulos Theodore Papatheodorou  
High Performance Information Systems Laboratory (HPCLab)  
Computer Engineering and Informatics Dpt.  
University of Patras  
Patras-Rio, Greece  
[kaloukat@ceid.upatras.gr](mailto:kaloukat@ceid.upatras.gr), {pomonis, kotsomit, tsp}@hpclab.ceid.upatras.gr

**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. Towards this perspective, in this work we introduce an application based on a 3-tier architecture that illustrates the potential for combining Web 2.0 and Semantic Web technologies. This application consists a framework for searching books from Amazon and Half EBay. The implementation's backbone is focused on developing the underlying ontology, writing a set of rules for personalisation and creating a mashup with the use of Web APIs.

*Semantic Web; knowledge representation; ontology; rules; SWRL; personalisation; Web 2.0; mashups; Web APIs; Amazon API, eBay API, Web 3.0, 3-tier architecture*

## I. INTRODUCTION

The Semantic Web and Web 2.0 are two visions that seem to 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, 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. Recently, the term Web 3.0 came to be added in the glossary of Web and seems to describe the long-term future of the Web. 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, pulls in user profiles and identities, and must be combined for Web 3.0 to work [6]. Consequently, the integration of Semantic Web and Web 2.0 principles will conduce to the development of Web 3.0.

Towards this direction, in this work we attempt to build a web application based on a 3-tier architecture (proposed by [11]) which combines the basic principles of Semantic Web and Web 2.0, as mentioned above. We call this application *Books@HPCLab* and in the remainder of this paper, we refer to this application with this name. The users of this application have the ability to perform book searching and retrieve

metadata for books which fit their personal preferences, from different data sources such as Amazon and Half eBay. Each user constitutes an autonomous entity for the application and makes a profile with his interests and preferences. As a result, the content of the application is adapted to the profile of the user.

The implementation of 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. The application does not interact with a database, as it is usual, but presents content that it retrieves from an ontology. This makes the application's presented information more reusable and more effectively sharable. Application's content constitutes a data collection from different and heterogeneous sources of the Web. The term *mashup* is used to describe this heterogeneous combination of data and can be considered to have an active role in the evolution of Web 2.0. A main characteristic of Books@HPCLab application is users' personalization, in the way described above, which is implemented with the use of *rules*.

The rest of this paper is organized in seven sections. In section II we start by providing some broad definitions and discuss the concepts of Semantic Web and Web 2.0. Furthermore, we discuss related work and the theoretical background of the research area. In section III, we describe in detail the proposed application, its architecture and its design. In section IV, we explain the process of collecting data from the various data sources. In Section V, we outline some indicative application running examples in order to illustrate the features of the proposed application. Finally, we discuss future work and summarize our conclusions.

## II. BACKGROUND

The Semantic Web<sup>1</sup> 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, the Web content will be presented in a form that is more easily machine-understandable, which it means that machines will become much better able to process, to "understand" and to integrate the information that they simply display at present [5]. At the core of the Semantic Web

---

<sup>1</sup> <http://www.w3.org/>

architecture [5], appears *reasoning*, the key component for the derivation of facts expressed unexplicitly in an ontology.

All these Semantic Web technologies constitute an environment capable of enabling efficient and personalised applications, since the idea of personalisation is embedded with the very nature of Semantic Web and set the state of art. More precisely, the members of the working group A3, in the REWERSE project, investigate personalisation approaches on the Semantic Web and result in three personalisation techniques based on (i) reasoning about actions, (ii) transforming and adopting adaptive hypermedia techniques, and (iii) rule-based user/learner modelling [1,4]. In the context of the Books@HPClab, we have adopted the third proposed technique in a more simplified way.

Note that many ontology languages have restrictions of their expressiveness for the sake of decidability. One way to address this problem is to extend these languages with some form of rule languages. *Rules* have practical implementation in many domains, such as Engineering, Commerce, Law, Medicine, Internet and so on. Especially for the OWL, the extension of OWL DL with Horn-like rules gives an extended language, named SWRL (Semantic Web Rule Language)<sup>2</sup>, which is intended to be the rule language of Semantic Web [9,2]. The unrestricted combination of formalisms conduces to a very expressive formalism, which is at the same time, unsurprisingly undecidable. To overcome this risk, a *safety* condition is imposed on SWRL rules. This safety condition is known as “*DL-safety*” and such rules are called “*DL-safe SWRL rules*” or “*DL-safe SWRL rules*” and are used in the context of our work.

On the other side, the supposed competing vision for the future of Web, the Web 2.0 [6,8,11], which converts users from “*content consumers*” to “*content producers*”, is expressed in our application by the notion of mashup. A mashup is a hybrid web application, which combines heterogeneous data or functionality from two or many more external sources in order to create a new, unified and enhanced service or site. Web APIs (Web Application Interface), Web Feeds (RSS/Atom) and Screen Scraping are the main methods, which are used commonly for mashups’ creation, although the most usually used method is the first, Web APIs, and this which is used in our work. The development of mashups is full and anodic and many kinds of mashup appear continuously, as mentioned in [12].

The concept of combination Semantic Web technologies and Web 2.0 technologies is not our invention, a number of recent papers have investigated the topic from different angles. Towards the combination of Mashups with the notion of Semantic Web [3], many attempts have been made such as, (i) Semantic Mashup for Tourism, (ii) Usual map mashups are converted to Semantic Map Mashups, (iii) Semantic Mashups for several scenarios in the life sciences, (iv) the use of Mashup Architecture in more sophisticated tasks, like business processes.

However, our approach takes these combination paradigms a step further by augmenting the user experience with a set of SWRL personalisation rules.

### III. SYSTEM ARCHITECTURE AND DESIGN

As illustrated in the following figure, the layers of the architecture could be distributed both at logical and physical level: (i) the front-end layer, (ii) the application logic layer and (iii) the knowledge management layer. It is also worth noting that each of the three layers may be physically located on different computer systems. This is an adapted version of the architecture proposed in [12] which seems to better fulfil our needs.

#### A. BookShop Ontology

In the lower part of the adapted 3-tier architecture, named *Knowledge Management layer*, there is the knowledge base of the entire application, which means the core ontology, *BookShop*, with personalisation rules and the individuals (instances of classes).

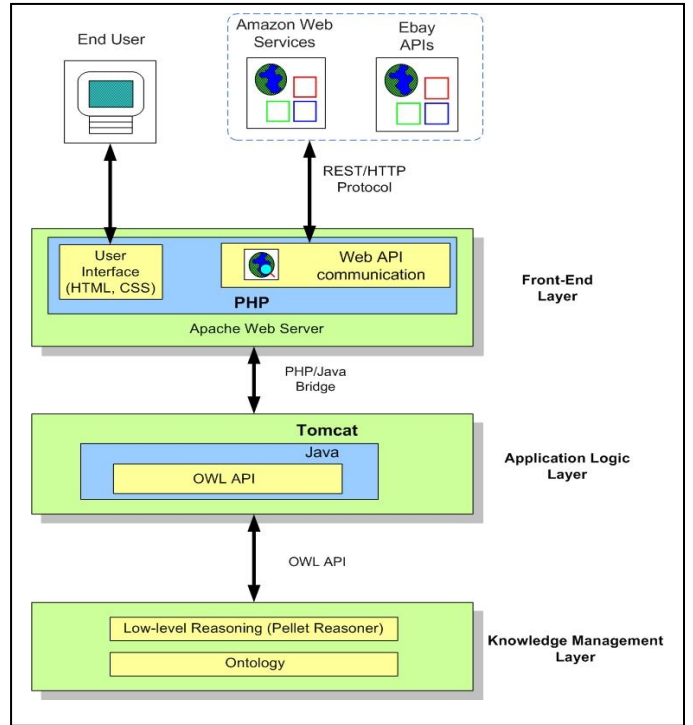


Figure 1. The adapted 3-tier architecture

Although, there are many formal methodologies for developing ontologies [7] (TOVE, KACTUS, METHONTOLOGY, On-To-Knowledge, etc), we preferred a more simplified and intuitive approach for this purpose [10]. In short, it is an iterative approach to ontology development, starting with a rough first pass at the ontology, then revising and refining the evolving ontology and filling in the details. This iterative designing process will likely continue through the entire lifecycle of the ontology. Our designing attempt is resulted in the core ontology BookShop and part of this is represented in figure 2.

In a brief analysis of our application, a registered *User* visits our web application, Books@HPClab, in order to retrieve

<sup>2</sup> <http://www.w3.org/submission/SWRL>

metadata for *Books* which match his profile. A *User* is assumed to be registered in our application only when he disposes password and username. User's profile consists of four fields (user's personal preferences): (i) the preferable Condition, (ii) the preferable Rating, (iii) the preferable Publication Year and (iv) the preferable maximum Price, which user's favourite books must have. The available information for each book, gathered by Amazon and Half EBay as illustrated in figure 1, include the surname and the firstName of *Book's Author*, the Book's title, publisher, dimensions, ISBN, publication year, number of pages, format, rating. Usually, images in various sizes and a url to visit the online bookstore of Amazon are also available for the user. In addition, a synopsis of the purchase *Offers* from Amazon and Half EBay is also made available. For each *Offer*, the Condition, the Price and the Origin are presented.

From the above paragraph, it becomes clear that the words written in *italics* are the main concepts of the BookShop ontology. These four concepts are represented by the four classes of the ontology (Figure 2): Book, Offer, User and Author. Note also that the class Author is subclass of the class Person, which is included in the FOAF (Friend of a friend) ontology. FOAF is an ontology which provides a unified way to describe persons, expressing their interests, their activities and their relations to other people and objects. We select the class Person and especially, the properties *foaf:surname* and *foaf:firstName*.

All the underlined words, in the brief analysis, represent most of datatype properties of the BookShop ontology. Datatype properties link individuals to data values. The dashed arrows, showed in Figure 2, represent the datatype properties.

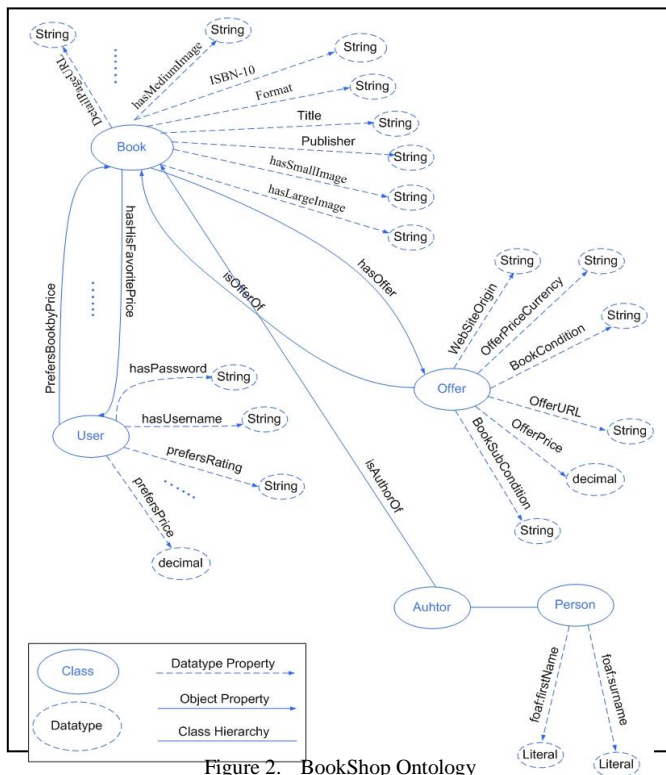


Figure 2. BookShop Ontology

On the other hand, the solid arrows represent the object properties of the ontology, which 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*). There is also a number of object properties that link an instance of Book to an instance of User and vice-versa. These properties express the user's preference for a book, depending on which preference fields in the user's profile are matched. For example, the object property *prefersBook\_byRate2\_Con\_Date* represents the preference relation between a User and a Book. This relation holds exactly when the book exhibits the desirable condition and publication year (i.e. two criteria are matched). In case when more criteria or different combinations of them are matched, then other relations hold, for example *prefersBook\_byRate2\_Con\_Rating*, which holds when the book has the preferred condition and rating, and *prefersBook\_byRate3\_Con\_Date\_Rating* which holds when the book has the preferred condition, publication year and rating.

### B. Rules for user personalisation

In order to reflect user personalisation in the ontology, a set of DL-safe rules was written, using the SWRL. These rules "match" user's preferences (user profile) with the features of books, which are returned as a result from searching Amazon and Half EBay web services. The "personalisation rules" distinguish those books which satisfy user's preferences from the entire set of books after the searching process. Initially four rules were written to check the satisfiability of each preference's criteria separately. Take, for example, the case where, in the underlying ontology, there is a rule about Book preferred Condition (for example New, Used etc). The SWRL description for this rule would be:

```
Book(?y), Offer(?z), User(?x), isOfferOf(?z, ?y), BookCondition(?z, ?condition),
prefersCondition(?x, ?preferred_condition), equal(?condition, ?preferred_condition) → prefersBookbyCondition(?x, ?y)
```

Figure 3. Rule for Condition of book

In case where two or three or four preference criteria are satisfied together, we wrote more rules so that to check the number of satisfiable criteria and cover all possible combinations. An example of such a rule follows:

```
Book(?y), User(?x), prefersBookbyCondition(?x, ?y),
prefersBookbyPrice(?x, ?y), prefersBookbyPublicationDate(?x, ?y) →
prefersBook_byRate3_Con_Price_Date(?x, ?y)
```

Figure 4. Rule for three preference criteria

We use Pellet as the underlying reasoner of our application which processes the ontology and fires the aforementioned rules.

## IV. COLLECTING DATA FROM BOOKSTORES

In this section, we review the process of searching information about books from the web data sources, in other words application's interaction with Amazon and Half EBay Web APIs. 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, among the set of available ECS operations, we choose the *ItemSearch* operation.

The ECS, like all AWS, is a REST-based API that requests are encoded into a simple URL string and responses are delivered as XML files. An example of a request follows:

```
http://ecs.amazonaws.com/onca/xml?Service=AWSECommerceService&
AWSAccessKeyId=XXXXXXXXXX&Operation=ItemSearch&SearchInd
ex=Books&keywords=Semantic%20D&ResponseGroup=ItemAttributes,
Offers,Images,Reviews,BrowseNodes,Offers&ItemPage=1&MerchantId
=All&Condition=All&Version=2009-01-06&Timestamp=2009-01-
01T12:00:00Z
```

Figure 5. Amazon request

In order to increase the accuracy of the search results, we set values for some *ItemSearch* input parameters in the URL request. For example, the setting value for *SearchIndex* parameter is always “Books”. Everything that is inserted in the search form of the application by the user is passed as value for the *keyword* parameter. The values set for the *ItemPage* parameter start from 1 up to the last page of results. Finally, for the *ResponseGroup* parameter, which controls the kind of information returned by the request, we set the following values: *ItemAttributes*, *Images*, *Offers*, *Reviews* and *BrowseNodes*.

A request 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 combine all these files in one unique XML file using DOM XML in PHP. This file is further processed by an XSLT in order to rule out redundant data that are not necessary in our implementation.

Once our application completes the search process at Amazon, it starts searching Half Ebay. We use the *eBay Shopping Web Services* and particularly, the *FindHalfProducts* operation. The interaction with the eBay Shopping API, like ECS of AWS, is based on the REST-protocol and the exchange of URL requests and XML files-responses.

The search at Half EBay would be similar to this at Amazon, based on keywords, but in this case, Shopping API returns too few results or returns error messages asking for a definition of a more specific request. The chosen way to address this problem was to direct the search at Half EBay by the Amazon’s results. In short, we search at Half EBay based on the ISBN of each book returned as result by Amazon. So, we sent as many requests to Shopping API as the number of books returned by the ECS. The form of such “*Half EBay requests*” is the following:

```
http://open.api.ebay.com/shopping?callname=FindHalfProducts&respons
encoding=XML&appid=HPCLABd29-daf7-4f27-be86-
777875a6d01&version=603&ProductID.type=ISBN&ProductID.Value=
value_of_isbn&IncludeSelector=Items
```

Figure 6. Half EBay request

When the searching process at Half EBay is complete, data are collecting in one unique XML file. Then, the elements of this XML file are converted into individuals of the BookShop ontology by means of another XSLT in OWL form. A

fragment of the final XML file and the corresponding OWL file are shown below:

```
.....
<Author>Toby Segaran</Author>
.....
```

Figure 7. XML description of an Author

```
.....
<Author rdf:ID="Author_id2134438">
  <foaf:firstName xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  rdfs:datatype="http://www.w3.org/2001/XMLSchema#Literal">Toby</f
  oaf:firstName>
  <foaf:surname xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  rdfs:datatype="http://www.w3.org/2001/XMLSchema#Literal">Segaran
  </foaf:surname>
  <isAuthorOf rdf:resource="#Book1"/>
</Author>
.....
```

Figure 8. OWL description of an Author

A diagram, which shows the complete data flow, when application interacting Web APIs, appears in the following figure:

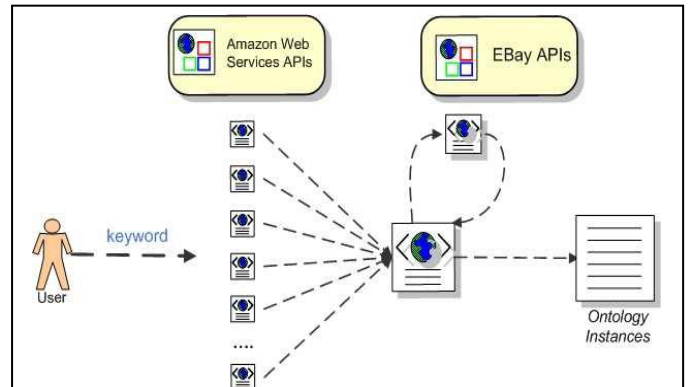


Figure 9. Diagram of communication with Web APIs

## V. FUNCTIONALITY AND EXAMPLES

In this section, we outline some indicative examples of the application in order to point out its capabilities, its features and its functionality.

The first page of the application includes two choices for the visitor, “*New User*” and “*Registered User*”. The first choice concerns users which never have visited the application before. Checking this choice, the user goes in another page, which includes a completion form of new user’s information such as username, password. This form includes also fields such as *Book Condition*, *Maximum Book Price*, *Publication Year* and *Maximum Book Rating* in order to determine his individual profile.

On the other hand, checking the second choice “*Registered User*”, the user goes to a page with a simple login form.

When a registered user enters his login account (username and password) correctly or when a new user is registered successfully in the application, the user then is directed to the main page, which includes a search form. This form consists only of a “Search” button and a textfield, where the user types a keyword or a keyphrase for book searching. When this button is pressed, the process, described at the previous section (*Collecting Data From Bookstores*), gets executed. Besides the search form, a menu of two choices is shown in right of the page. Hitting the *Settings* link, the user is directed to a form displaying personal settings for his profile and may modify his preferences, if he wants it. The *Logout* link signs the user off his account on the application and redirects him to the login form, described above.

When searching ends, results are imported into the core ontology (BookShop) as individuals. 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 ranked based on the rules outcome, as shown in Figure 10. In particular, the more criteria a Book satisfies (the more preference rules it triggers), the higher it appears in the results.

α/α	Τίτλος Βιβλίου	Score
1	Programming the Semantic Web	★★★★
2	Semantic Web For Dummies (For Dummies (Computer Tech))	★★★★
3	Semantic Web Programming	★★★★
4	Spiriting the Semantic Web: Bringing the World Wide Web to Its Full Potential	★★★★
5	A Semantic Web Primer (Cooperative Information Systems)	★★★★
6	Thinking on the Web: Browsers, Lee, Godel and Turing	★★★★

Figure 10. Presentation of preferred books for User\_1

Each table's row includes information, such as an auto-incremented number, the book's title and a number of exclamation marks which express the number of satisfied criteria. Book's title is a link and clicking it, a pop-up window appears with all the available features of the specific book, as shown below.

α/α	Τύπος	Κριτήριο	Web Origin
1	New	18.45	amazon.com
2	New	18.5	ebay
3	New	34.94	ebay
4	New	13.97	ebay

Figure 11. Pop-Window

This pop-up window presents general information about the selected book, such as Title, Author, ISBN etc are presented. The Title and the image of book are links to “official” book page of Amazon. Finally, the Offers for this book are presented in a form of table. Following the *Search* link (see Fig.11), user returns back to the page with the search form in order to initiate a new search.

## VI. FUTURE WORK

Regarding future work, our application can benefit from implementation improvements and research that can be summarized in the following points:

- Test our application in real conditions with significant load in order to demonstrate and evaluate its features and potentials.
- Response time of our application is susceptible to improvement. The small observed delay is caused by the use of the DOM parser when reading and retrieving data from the XML files.
- The BookShop core ontology follows OWL DL, which could be replaced by OWL 2, since it is the newest version of the OWL language.
- Data sources for mashup creation could be increased not only in number but also in type (various websites, Web APIs, RSSfeeds etc).
- Make this pilot application available and more known to the public in order to get feedback especially from other researchers and web developers and modify or enrich it.

## VII. CONCLUSIONS

In this work, we have shown that Semantic Web and Web 2.0 can be complementary visions for the future of 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 present a prototype web application, which integrates information from Web APIs, such as Amazon Web API and Half EBay API and converts them to individuals of an ontology schema, finally resulting in a kind of a semantic mashup for books. This semantic mashup is accompanied by a set of rules, which impose a high level of personalisation. It is our belief and hope that this work is a step towards the evolutionary process of Web, and will inspire other researchers and common users to support and deal with the idea of Web 3.0.

## REFERENCES

- [1] Grigoris Antoniou, Matteo Baldoni, Cristina Baroglio, Robert Baumgartner, Francois Bry, Nicola Henze, Wolfgang May, Viviana Patti, Slawomir T. Wierzbichon. “Personalization for the Semantic Web II”. A3-D4, REVERSE, August 2005

- [2] Grigoris Antoniou, Carlos Viegas Damásio, Benjamin Grosf, Ian Horrocks, Michael Kifer, Jan Maluszynski, Peter F. Patel-Schneider. "Combining Rules and Ontologies". A survey. I3-D3, REWERSE, 2005
- [3] Anupriya Ankolekar, Markus Krötzsch, Thanh Tran, Denny Vrandečić. "The two cultures: Mashing up Web 2.0 and the Semantic Web". Web Semantics: Science, Services and Agents on the World Wide Web, Semantic Web and Web 2.0, Volume 6, Issue 1, Pages 70-75, February 2008.
- [4] Matteo Baldoni, Cristina Baroglio, and Nicola Henze. "Personalization for the Semantic Web". A3-D7, REWERSE, September 2006
- [5] Tim Berners-Lee, James Hendler, Ora Lassila. "The Semantic Web". Scientific American Magazine, May 2001.
- [6] Allan Cho (Jul 22, 2008). "What is Web 3.0? The Next Generation Web: Search Context for Online Information". [http://internet.suite101.com/article.cfm/what\\_is\\_web\\_30](http://internet.suite101.com/article.cfm/what_is_web_30)
- [7] Oscar Corcho, Mariano Fernández-López, Asunción Gómez-Pérez. (November 2001) "Methodologies, tools and languages for building ontologies. Where is their meeting point?". Data & Knowledge Engineering, Volume 46, Issue 1, Pages 41-64, July 2003.
- [8] Luciano Floridi. "Web 2.0 vs. the Semantic Web: A Philosophical Assessment". University of Hertfordshire; Faculty of Philosophy and IEG, University of Oxford. Address for correspondence: Department of Philosophy, University of Hertfordshire, de Havilland Campus, Hatfield, Hertfordshire AL10 9AB, UK; [l.floridi@herts.ac.uk](mailto:l.floridi@herts.ac.uk)
- [9] Pascal Hitzler and Bijan Parsia. "Ontologies and Rules". In Steffen Staab and Rudi Studer (Ed.), *Handbook on Ontologies*, pp. 111-132, Springer 2nd Edition, 2009.
- [10] Natalya F. Noy, Deborah L. McGuinness. "Ontology Development 101 : A Guide to Creating Your First Ontology". Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.
- [11] Tim O' Reilly. "What Is Web 2.0, Design Patterns and Business Models for the Next Generation Software". International Journal of Digital Economics 65 pp. 17-37. November 2007.
- [12] Tzanetos Pomonis, Dimitrios A. Koutsomitropoulos, Sotiris P. Christodoulou and Theodore S. Papatheodorou. Towards Web 3.0: A Unifying Architecture for Next Generation Web Applications. In S. Murugesan (Ed.), *Handbook of Research on Web 2.0, 3.0 and X.0: Technologies, Business and Social Applications*, pp. 192-204. IGI Global, 2009.