Chapter

# Towards Web 3.0: A Unifying Architecture for Next Generation Web Applications

**Tzanetos Pomonis**
*University of Patras, Greece*

**Dimitrios A. Koutsomitropoulos**
*University of Patras, Greece*

**Sotiris P. Christodoulou**
*University of Patras, Greece*

**Theodore S. Papatheodorou**
*University of Patras, Greece*

## ABSTRACT

While Web 2.0 term is used to describe the current trend in the use of web technologies, Web 3.0 term is used to describe the next generation web, which will combine Semantic Web technologies, Web 2.0 principles and artificial intelligence. Towards this perspective, in this work we introduce a 3-tier architecture for web applications that will fit into the Web 3.0 definition. We present the fundamental features of this architecture, its components and their interaction, as well as the current technological limitations. Furthermore, some indicative application scenarios are outlined in order to illustrate the features of the proposed architecture. The aim of this architecture is to be a step towards supporting the development of intelligent semantic web applications of the near future as well as supporting the user collaboration and community-driven evolution of these applications.

## INTRODUCTION

Current trends in Web research and development seem to revolve around two major technological pillars: Social-driven applications, a main component in the Web 2.0 domain, and the Semantic Web. It is our firm belief that web semantics and Web 2.0 are complementary visions about the near future of the Web, rather than in competition: surely they can learn from each other in order to overcome their drawbacks, in a way that enables forthcoming web applications to combine Web 2.0 principles, especially those that focus on usability, community and collaboration, with the powerful Semantic Web infrastructure, which facilitates the information sharing among applications. Recently, the term Web 3.0 is used to describe the long-term future of the web (Lassila, 2007; Hendler, 2008). Web 3.0 will surely incorporate semantic web and Web 2.0 principles, but researchers believe that it will also include some more sophisticated concepts like artificial intelligence on the web.

Towards this direction, in this work we propose a 3-tier architecture for web applications that will fit into the Web 3.0, the next generation web. At the lower layer of the architecture, we introduce and describe an advanced semantic knowledge base infrastructure that can support integration of multiple disparate data sources, without requiring a concrete underlying semantic structure. In addition, the upper layers of the architecture provide greater flexibility in the user interactions with the underlying ontological data model. As a result, it supports user collaboration and community-driven evolution of the next generation web applications.

This architecture gives the developers the ability to build complicated web applications which combine the philosophy of Web 2.0 applications, and the powerful technical infrastructure of the Semantic Web, supported by applying Artificial Intelligence principles on the Web. Furthermore, this architecture is well suited for supporting enhanced Knowledge Systems with advanced knowledge discovery characteristics, towards the future implementation of an Internet-scale Knowledge System. For example, the proposed architecture could be used to enrich current wiki applications towards next generation semantic wiki platforms that will mash-up scattered data sources and provide intelligent search capabilities.

The following text is organized in five sections. In section 2 we start by providing some broad definitions and discussing the concepts of Semantic Web and Web 2.0. Furthermore, we discuss related work and the theoretical background of the research area. In section 3, we describe in detail the proposed architecture, its components, its fundamental features and the current technological limitations. In section 4, we outline some indicative application scenarios in order to illustrate the features of the proposed architecture and prove that it can be applied today and support modern web applications. Finally, we discuss future work and summarize our conclusions.

## BACKGROUND

As Semantic Web and Web 2.0 were firstly introduced separately by groups with completely contrary beliefs on the evolution of World Wide Web, and even targeting different audiences, there has been a common perception that both are competing approaches for organizing and emerging the Web.

The Semantic Web, outlined by Berners-Lee (2001), becomes a revolutionary technological approach for organizing and exchanging information in a cross-application dimension. Strongly supported by World Wide Web Consortium and powered by heavy academic and enterprise research, Semantic Web can demonstrate standardized and well-defined approaches in language description, such as RDF (Manola, 2004), RDF(S) (Brickley, 2004) and Web Ontology Language OWL (Smith, 2004), as well as research background in ontology engineering and modeling tools, from SHOE (Heflin, 1998) to Protégé (Knublauch, 2004).

Semantic Web is powered by a strong AI background through its foundation on the Description Logics (DL) formalism (Baader, 2007). DL languages have become in recent years a well-studied formalism, originating from Semantic Networks and Frames and, as such, they have been extensively used in formal Semantic Web specifications and tools.

These languages are of variable expressive strength which comes with the cost of increased computational complexity. Therefore, current research in this area is focused on efficient and advanced algorithms and procedures that would provide intelligent querying capabilities for the real word Web, based on DL descriptions and possibly subsets of and reductions from them that may exhibit more satisfying computational properties (Grau, 2008).

One main reason for transforming the current Web to a Semantic Web is the ability to deduce new, un-expressed information that is only implied by existing descriptions. If the Web is to be considered as a huge, distributed knowledge base, then well-known AI techniques, at least for the part with sound foundations in logic, can be utilized in order to form the basis for intelligent negotiation and discovery on the Semantic Web. Such techniques may include for example deductive query answering and inference-based reasoning (Luke, 1996; Berners-Lee, 2001).

On the other hand, the Web 2.0 term, introduced by Tim O'Reilly (2005), represents a widely spread trend of adopting certain technologies and approaches in web development, targeting more flexible and user friendly applications, and easier distributed collaboration. The usability aspect is met by Rich Internet Applications (RIA) (Loosley, 2006) and especially Asynchronous JavaScript and XML (AJAX), which support the creation of responsive user interfaces as well as more interactive browsing experience. Collaboration conveniences come through the creation of virtual online communities of users that contribute effort and data to a common cause, achieving better results than each individual could do on his own. Finally there is a greater flexibility in data handling, enabling the development of hybrid web applications, called Mash-ups, which combine discrete data sources and services from different sites in order to provide a unified and enriched result.

Therefore, the Semantic Web can provide a rich and powerful technical infrastructure for any kind of web application, while the paradigm of Web 2.0 applications can be used to provide useful guidelines, focusing on usability and collaboration. Thus, the Semantic Web and Web 2.0 principles can be combined as complementary approaches to provide more efficient web applications. Such applications could be thought to be part of next generation's web and seem to fall under the term *Web 3.0* (Hendler, 2008), which lately is sort of "talk of the town" (Lassila, 2007).

In this context, there are several approaches; from developing AJAX tools for the Semantic Web (Oren, 2006) and studying the combination of ontologies and taxonomies (Mika, 2005), up to the proposition of sophisticated hybrid architectures, combining both of these technologies (Ankolekar, 2007).

All of the above are of great use in any data-handling web application, and where there is need for a knowledge system. Especially for next generation knowledge systems that try to benefit from Web 2.0 approaches and collaborative development in order to build, or more precisely grow, Internet-scale knowledge systems (Tenenbaum, 2006).

## PROPOSED ARCHITECTURE

In this section we propose an architecture for web applications, which provides developers the ability to structure complicated web applications, which combine the vision of Web 2.0 and the rich technical infrastructure of the Semantic Web, supported by applying Artificial Intelligence principles. Such applications could be next generation semantic wikis, intelligent mash-ups, semantic portals and in general any data-handling web application that intends to provide semantic information combined with advanced intelligent querying capabilities.

The information of these applications could be delivered by two main ways:

i. Directly to end users through the web-based interface of a stand-alone application
ii. To other programs or services, that act as intermediaries with third-party web applications, by interacting with the API of our semantic infrastructure to retrieve precisely the information they need.

A conformant implementation may follow the traditional 3-tier model, which lately (Hendler, 2008) is commonly used to support web 3.0 applications, with an important variation: Where a database server would be typically used, we now use a knowledge base system, since a traditional DBMS lacks the necessary features and functions for managing and utilizing ontological knowledge. Note that each of the three layers may be physically located on different computer systems. The proposed architecture is presented in figure 1.

In fact, from the designer's point of view, our architecture could be decentralized in at least two ways:

i. The Semantic Web knowledge bases that data is extracted from, could be both logically and physically distributed (in the case of OWL, this can be accommodated by the `owl:import` directive) and in such case, an application has to provide for their integration. This is necessary, since Web Ontologies are expected and already tend to be developed in parts and fragments, each addressing a specific view of knowledge. Therefore, it is evident that their combination and alignment could provide richer descriptions and more powerful inferences.

ii. The layers of the application could also be distributed both at the logical and the physical level: the front-end layer, the application logic layer and the knowledge management layer.

Such a truly decentralized architecture, in accordance to the traditional 3-tier paradigm, is not yet possible with the majority of the current state-of-the-art and highly expressive inference engines, due to limitations of their interface capabilities, as described later. On the other hand, such an approach can have a more substantial contribution to the utilization of semantic information by users and applications by eliciting more obvious value from ontological data (Hendler, 2008).

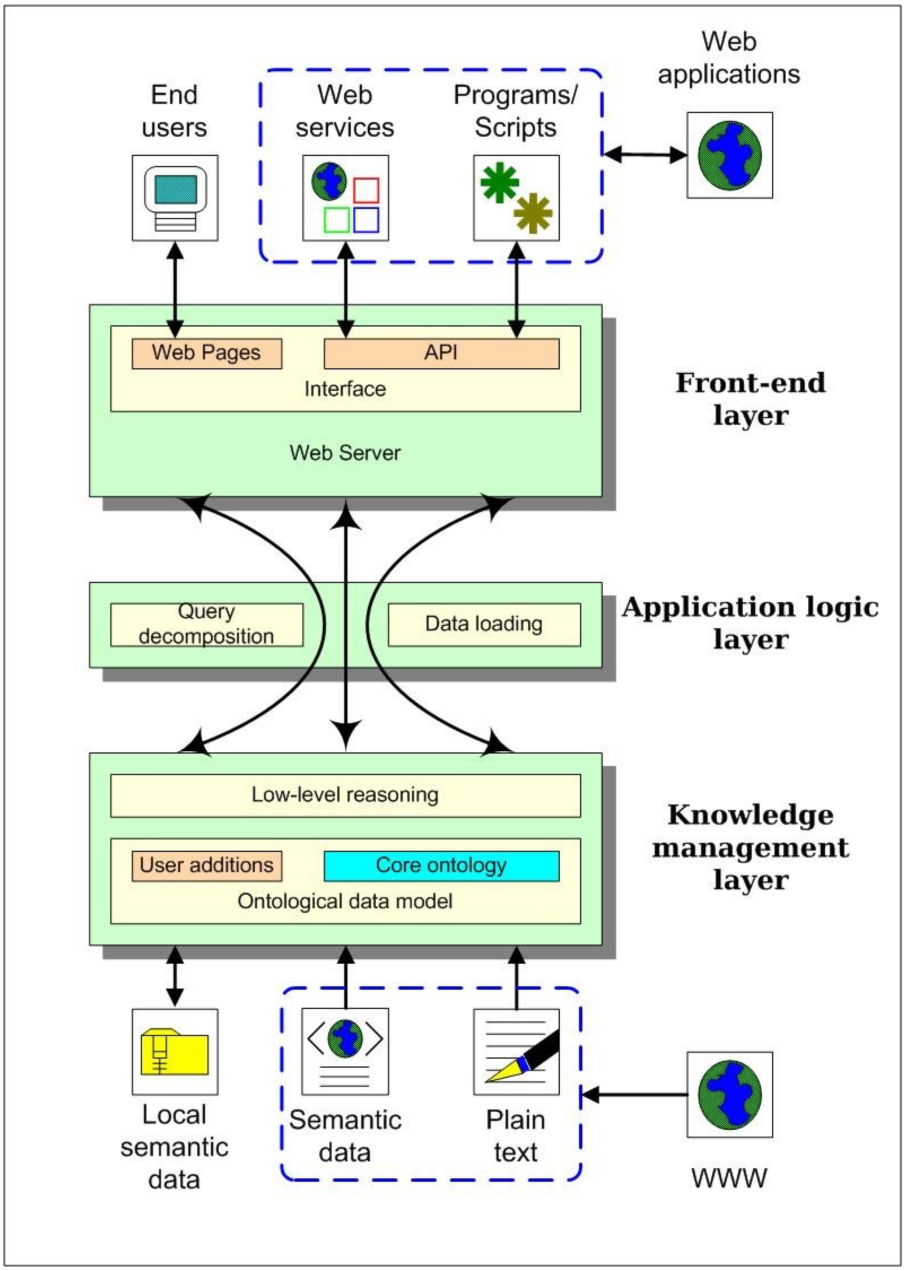


*Figure 1: The proposed 3-tier architecture*

The lower part of the proposed 3-tier architecture is a **knowledge management** layer (or system) which integrates and administers data sources that may be disparate in nature: ontology documents, metadata, feeds and other information with underlying semantic structure of variable density, from *semantic data* to *plain text* (zero density). As a result, this layer acts as a semantic mash-up that aligns information to a common, mediating ontology (the *core ontology*); at the same time this layer performs the *low-level reasoning* functions that are required in order to deduce implied information. Such an implementation can load Semantic Web Knowledge bases (OWL documents) that are available either on the local file system, or on the Internet. A temporary copy of every document is stored locally and is then loaded by the knowledge base server (an inference engine like RACER). RACER (Haarslev, 2003) can create and store in memory an internal model for each ontology it classifies. Classification takes place once for each ontology, during its initial loading.

User requests, queries, additions and other interventions to the ontological model are being interpreted through the **application logic** layer. This is responsible for the *ontological information loading*, proper rendering / presentation of it to the user and the *decomposition of the user requests* to low-level functions of the knowledge management system. Ontological data and reasoning results (Koutsomitropoulos, 2005) are fetched by interacting with knowledge management system, which could physically be located in another machine, e.g. over the TCP/IP protocol. In case of using RACER, this interaction is greatly facilitated through the JRacer API. The application logic can be implemented using the Java programming language, as well as JSP, JavaBeans and Java Servlets technologies. Tomcat can be used as an application server.

Individual users or users being part of communities may interact with the underlying knowledge base through the **front-end** layer of the architecture on a reciprocal basis: this means that they are not confined to the mere ingestion of data sources; rather, they are also enabled to fully interact with them, by adding, commenting and incrementing on the underlying *ontological data model* (*user additions*). On a standalone web application scenario, this is accommodated through web pages, either static ((X)HTML), or dynamic implementing rich interfaces, where, for example, the user experience is enhanced by the AJAX paradigm in JSP pages rendered by the browser. However, *web services*, *programs/scripts* and other interoperability interfaces may also interact as clients with the front-end layer. Communication with the application layer can be conducted over the HTTP protocol, using forms.

## Semantic mash-up

A conformant application should be able to handle information originating from a number of sources and organized with different levels of semantic density.

For the purpose of our work, semantic density of ontology-forming information can be defined as the extent to which the intended models of the ontology can capture the domain conceptualization. This definition is consistent to the definition of ontology, as introduced by Guarino (1998).

Roughly, we can distinguish among three types of such information:

i. Information that is already adequately described and its semantics expressed in machine readable ways. Ideally, this kind of information is serialized in web ontology languages, such as OWL (or it is trivial to do so).
ii. Information that is organized as a flat aggregation of annotations, as it is most often the case with metadata schemata. Such a schema may imply an underlying semantic model but this is not adequately captured. However each annotation has distinguished semantic interpretation from each other.
iii. Information that is being given as simple, unorganized text, in the form of natural language. Semantics that may be hidden in such descriptions are not expressed in any way.

In the first two cases, we can employ a technique known as semantic profiling (Koutsomitropoulos,

2007) in order to intensify the semantic density of information. This in turn would increase the expressivity of descriptions leading to the ability to process and respond to more powerful, inference-based queries. Even in the case where annotations are flatly organized as metadata elements, we can construct a fully-structured ontology model out of them, enriched with new constructs specific to our application or constructs that capture relations already implied in the schema; then, we can align available descriptions to our new ontology by using an automated translation process (e.g. based on XSLT), requiring no end-user intervention.

The third case is the trickiest one, since it offers no starting point to be based upon. Necessarily, it would require some form of natural language processing (NLP) (Alani, 2003) in order to identify, for example, keywords that may reveal the subject classification of the textual description. Such keywords can then be mapped to an existing ontology, such as WordNet (http://wordnet.princeton.edu/), in order to extract semantic relations among them and populate, to a limited extend, our common mediating ontology.

## Advanced interconnectivity features

The front-end layer of the proposed architecture can support stand-alone web applications that provide an enhanced user experience which is accommodated through rich interfaces. Targeting a Rich Internet Application (RIA) (Loosley, 2006), where a web application has the features and functionality of traditional desktop applications, using advanced Web 2.0 approaches, like the AJAX technique, where the necessary processing for the user interface is typically transferred to the web client, but the bulk of the data is kept back on the application server.

However, there is prediction for additional interconnection features. A variety of interoperability interfaces may also interact as clients with the front-end layer of the architecture. For example, a conformant web application can facilitate third-party developers integrating its freely distributed semantic information into their web sites, by providing direct, high-level access to the data of its knowledge base, through its API.

Third-party web services can reach a high level of interoperability through the architecture's API to provide interconnection to third-party web applications using web service specific techniques, e.g. communicating using XML messages that follow the SOAP standard. In addition, programs or scripts (written in any language) that conform to the API of our architecture can have access to the data of the knowledge base. Finally, a third-party developer can use web scraping techniques to extract content from any website over HTTP for the purpose of transforming that content into another format suitable for use in his web application.

## Community interaction

Collaboration conveniences are essential features of this architecture. In order to achieve better results in growing and supporting a conformant application, users should be allowed to contribute effort and data**.**

In this way user information can contribute to the population of the application's ontology schema. There may be cases however, where the alternation of the ontological schema itself may be desirable. For example, administrators and power users should be able to define new ontology classes or properties and these definitions are incorporated or imported in the central ontology. Of course such alternations are to be done in an incremental-only way, since the knowledge on the Semantic Web is inherently monotonic. Moreover, one has to be careful making these additions, in order to avoid redundancy, i.e. multiple equivalent descriptions that are being repeated. To this end, the frequent classification and consistency checks on the ontology may be helpful, since completely identical descriptions can be identified through reasoning.

## Technological limitations

The best choice for the underlying formalism for our methodology is to use at least OWL DL, since this OWL dialect offers a satisfactory expressivity level, adequate to powerful inferences (Horrocks, 2003). However, the majority of the current state-of-the-art and highly expressive inference engines lack in fully supporting the specific requirements of our architecture.

FaCT++ and Pellet are currently the only two DL-based engines that appear to fully support the decidable subset of OWL. However they only support DIG 1.1, which is insufficient for full OWL DL support (Dickinson, 2004), a fact that mostly drives the upcoming 2.0 specification.

DIG 1.1 communication takes place over HTTP and there is no other TCP/IP-like connectivity support; in the case where a tool or application needs to utilize these reasoners, one may use a programmatic API (e.g. Jena or the Manchester API) that interfaces these reasoners as direct in-memory implementations (Horridge, 2007). This approach may have the advantage of reducing the message-passing load of the DIG protocol, but surely is insufficient for developing truly decentralized Web applications and services for the Semantic Web. As DIG 2.0 specification that would solve the aforementioned problems is currently in flux, these reasoners cannot be used in developing a distributed web service for Semantic Web Knowledge Discovery that would fully support OWL DL.

In such a case we should opt for RACER as a DL-based reasoning back-end. RACER used to be dominant in terms of expressivity and interface abilities among DL-reasoners, when Pellet was not even existent. Now, RACER, being freely available for non-commercial purposes, is the only free engine, with expressive strength closest to OWL DL that exposes/maintains an independent, full-featured, IP-compatible communication interface.

# INDICATIVE APPLICATION SCENARIOS

In this section we outline some indicative application scenarios in order to illustrate the features of the proposed architecture and prove that it can be applied today and support modern web applications.

## Developing the application

Let's consider a web developer/engineer deciding to use our architecture, in order to develop and run a semantic wiki specializing in Cultural Heritage.

The first thing he has to do is to design the proper ontology, based on OWL, in order to completely describe the desired information that is to be presented through his site. This can be information about monuments, historical artifacts, ancient manuscripts, or even modern bibliography about cultural heritage. For this particular domain, a good starting point may be the CIDOC Conceptual Reference Model, a recent ISO standard (Crofts, 2003).

The next step is to decide whether he is going to use only locally created and stored information, as of a usual semantic wiki, or he is also going to gather information through the web. In the latter case he can search for sites with similar content and categorize them based on the density of their underlying semantic structure. Afterwards he has to map this information to his own ontology, either by using semantically enhanced application profiles (Koutsomitropoulos, 2007), for information with notable semantic structure, or by using natural language processing techniques, e.g. (Alani, 2003), for simple, unorganized text, like the one he can get from Wikipedia (http://www.wikepedia.org) articles.

Now he is ready to pick up the suitable software components for his conformant implementation of our architecture that will support his application. Such a combination could be: RACER as the inference engine, Tomcat as the application server and Java server-side technologies, e.g. JSP, JavaBeans and Java Servlets.

Finally, he develops a user-friendly and highly interactive front-end for his application, using for instance JSP supported by AJAX techniques. His main focus should remain to provide his end-users with web application modules (components) that make easier, not only making intelligent queries and entering new information in the wiki's knowledge base, but also fully interacting with data sources, by adding, commenting and incrementing on the underlying ontological data model.

As a result of the described bottom-up development procedure, the desired application is up and running. Based on the development infrastructure, the information mediated through this semantic wiki can be then collaboratively manipulated and enriched by its target users.

## Intelligent Querying

One of the main advantages of such a web application is to make possible for end-user to submit intelligent queries. Take, for example, the case where, in the underlying ontology, there is the expression that a sword is made of iron. The OWL description for this would be:

```
<owl:Class rdf:ID="sword">
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:onProperty rdf:resource="#isMadeOf"/>
    <owl:someValuesFrom rdf:resource="#Iron"/>
   </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

The expression describing that iron is metal would be:

```
<owl:Class rdf:ID="Iron">
  <rdfs:subClassOf rdf:resource="#Metal"/>
</owl:Class>
```

One can now retrieve every metal item, via the following expression.

```
<owl:Restriction>
   <owl:onProperty rdf:resource="#isMadeOf"/>
   <owl:someValuesFrom rdf:resource="#Metal"/>
</owl:Restriction>
```

A more complex example is the notion of “co-author”, which is of great use in the context of applications that host resources developed collaboratively (e.g. books, research papers in repositories, digital libraries items, wikis etc.). This particular relation in not often explicitly captured in metadata, for example the DC does not provide any field for this. A co-author relation, which is held among authors, is implied by author relations that exist between authors and items. In particular, consider an author A. This author is in “co-author” relationship with all other authors that are in “author” relationship with the items that A has authored.

This kind of relation is a typical example of the need for role-chains that are accommodated by OWL 1.1 (not even OWL DL). In Description Logics syntax (Baader, 2007), the notion of “co-author” can be described as:

$$\mathtt{author}^{-} \circ \mathtt{author} \sqsubseteq \mathtt{co\_author}$$

where $^{-}$ stands for inverse relation, $\circ$ for role composition, and $\sqsubseteq$ is the sub-property relation.
Notice also that this kind of relation cannot be (easily) described even in traditional DBMSs.

## Incrementing the ontological data model

Now let's imagine a user of the above semantic wiki. He has spent some time using the application and has become familiar enough with entering and editing content about the topics related to this cultural heritage wiki.

As he is really interested in the cultural domain, he notices that although this wiki is filled with a large amount of information about monuments, historical artifacts, manuscripts, and literature in general, it lacks specific information about paintings, although they are strongly considered to be a discrete field of interest in cultural heritage. Furthermore, he notices that the specific painting part was not taken into consideration during the initial design procedure of the application's underlying ontological data model, so he is not able to enter information about paintings in this wiki.

As he has now become an experienced user, sort of power user, of this application, he is aware of all its potential. This one is not a simple semantic wiki where users are confined to the mere insertion of data, but provides its users with advanced web application modules (components) in order to facilitate them to fully interact with its infrastructure. As a result, he decides to take advantage of this feature and enrich the underlying ontology, by including the required schema for painters.

For example he can define the class of “Painters” as a “Person” who has “performed” at least one “Painting_Event”. Note also that the class “Person” and the property “performed” refer to another ontological schema, namely CIDOC-CRM:

```
<owl:Class rdf:ID="Painter">
  <rdfs:subClassOf rdf:resource="&crm;E21.Person"/>
   <owl:equivalentClass>
    <owl:Restriction>
     <owl:onProperty rdf:resource="&crm;P14B.performed"/>
     <owl:someValuesFrom rdf:resource="#Painting_Event"/>
    </owl:Restriction>
   </owl:equivalentClass>
</owl:Class>
```

As a result this semantic wiki is now ready to receive information also for paintings and painters.

## Acting as a semantic proxy

While this semantic wiki works fine as a stand alone web application, which provides its users with comprehensive information for cultural heritage topics, it has also a lot to offer to other web applications.

This application is based on an open architecture, whose content and especially its underlying ontological data model are freely available, and therefore it could act as a proxy of semantically structured data. Thus every developer who wants to build an informational site for cultural heritage does not have to collect it over the Web and map it to a new ontology. All he has to do is to use this application's advanced interconnection features, e.g. its API, to have a unique repository of the desired information, and use it as is or further map this wiki's ontological model to his own one, a procedure which becomes trivial.

## Other indicative applications

Another indicative application could be a semantic movie portal. In such a portal, information for movies could be collected from Internet Movie Database (http://www.imdb.com) using ordinary web scraping techniques, while information about respective DVD releases could be collected from the Amazon website (http://www.amazon.com) using its API. All this information could be unified in a suitable ontology, creating a semantic mash-up. Thus, a common user of this portal, not only has all the information he needs in a single site, but additionally he could benefit from advanced features of semantic personalization (Tziviskou, 2007; Ankolekar, 2006) and intelligent querying support.

## FUTURE RESEARCH DIRECTIONS

Regarding the future work, it will include both implementations and research work that can be summarized in the following points:

- Specify, design and develop indicative web applications based on our architecture, in order to demonstrate, study and evaluate its features and potentials.
- Make these pilot web applications available and encourage users to participate, comment and enrich underlying ontological model. Study and evaluate the user collaborations and the community-driven evolution of the applications.
- Investigate analytically current web technologies in order to decide which ones are best fit into our architecture.
- Get feedback from other researchers and web developers on our proposed architecture and modify or enrich it.

## CONCLUSION

In this work we have shown that Semantic Web and Web 2.0 can be complementary visions about the future of the Web, rather than in competition. This was done by the proposition of a unifying architecture, which can be used to support any data-handling web application. Such applications could combine the philosophy of Web 2.0 applications, and the powerful technical infrastructure of the Semantic Web, supported by applying Artificial Intelligence principles on the Web. Applications with such features are considered to be the next generation web applications, or Web 3.0 applications.

Semantics and knowledge-discovery capabilities play a key role in this unifying architecture. We recognize, from a methodological point of view, reasoning and inferences as prominent features in Semantic Web scenarios that are necessary in order to enable intelligent services. Therefore, the lower part of the proposed 3-tier architecture is a *knowledge management layer*, where a database server is typically used in other architectures. This layer can support the integration of multiple disparate data sources, without requiring a concrete underlying semantic structure. User requests, queries, additions and other interventions to the ontological model are being interpreted through the *application logic layer.* Finally, the *front-end layer* of the architecture supports on one hand the rich interaction with users (and communities), and on the other hand the interoperability with other web applications through web services or other programs.

Overall, the proposed architecture is a step towards supporting the development of intelligent semantic web applications of the near future as well as supporting the user collaboration and community-driven evolution of these applications.

## REFERENCES

Alani, H., Kim, S., Millard, D. E., Weal, M. J., Hall, W., Lewis, P. H., & Shadbolt, N. R. (2003). Automated Ontology-Based Knowledge Extraction from Web Documents. *IEEE Intelligent Systems*, 18(1), 14-21.

Ankolekar, A., Krötzsch, M., Tran, T., & Vrandecic, D. (2007). The two cultures: mashing up

Web 2.0 and the Semantic Web. In *WWW '07: Proceedings of the 16th International Conference on World Wide Web* (pp. 825-834). New York, USA. ACM Press.

Ankolekar, A., & Vrandecic, D. (2006). Personalizing Web surfing with semantically enriched personal profiles. In M. Bouzid and N. Henze (Ed.), *Proc. Semantic Web Personalization Workshop*. Budva, Montenegro.

Baader, F., & Nutt, W. (2007). Basic Description Logics. In F. Baader, D. Calvanese, D. McGuinness, D. Nardi, & P. F. Patel-Schneider (Eds.), *The Description Logics Handbook* (2nd ed.). Cambridge.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American*, 5.

Brickley, D., & Guha, R. V. (2004). RDF Vocabulary Description Language 1.0: RDF Schema. *W3C Recommendation*. Retrieved 10 February 2004, from http://www.w3.org/TR/rdf-schema/.

Crofts, N., Doerr, M., & Gill, T. (2003). The CIDOC Conceptual Reference Model: A standard for communicating cultural contents, *Cultivate Interactive*, 9. http://www.cultivate-int.org/issue9/chios/

Dickinson, I. (2004). *Implementation experience with the DIG 1.1 specification* (Tech. Report HPL-2004-85). Hewlett Packard, Digital Media Sys. Labs, Bristol.

Grau, B. C., Motik, B., Wu, Z., Fokoue, A., & Lutz, C. (2008). OWL 2 Web Ontology Language: Profiles. *W3C Working Draft*. Retrieved from http://www.w3.org/TR/owl2-profiles/

Guarino, N. (1998). Formal Ontology and Information Systems. In N. Guarino (Ed.), Formal Ontology in Information Systems. *Proceedings of FOIS'98* (pp. 3-15). IOS Press.

Haarslev, V. & Möller, R. (2003). Racer: A Core Inference Engine for the Semantic Web. In *Proc. of the 2nd Int. Workshop on Evaluation of Ontology-based Tools (EON2003)*, pp. 27-36. Florida, USA.

Heflin, J., Hendler, J., & Luke, S. (1998). Reading Between the Lines: Using SHOE to Discover Implicit Knowledge from the Web. In *AI and Information Integration: Papers from the 1998 Workshop* (pp. 51-57). AAAI Press.

Hendler, J. (2008). Web 3.0: Chicken Farms on the Semantic Web. *Computer* 41(1), 106-108.

Horridge, M., Bechhofer, S., & Noppens, O. (2007). Igniting the OWL 1.1 Touch Paper: The OWL API. In *Proc. of the OWL Experiences and Directions Workshop (OWLED'07)*. Innsbruck, Austria.

Horrocks, I., Patel-Schneider, P. F., & van Harmelen, F. (2003). From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics,* 1(1), 7-26.

Knublauch, H., Fergerson, R. W., Noy, N. F., & Musen, M. A. (2004). The Protégé OWL plugin: An open development environment for Semantic Web applications. In *Proc. 3rd*

*International Semantic Web Conference (ISWC04)*. Springer.

Koutsomitropoulos, D. A., Fragakis, M. F., & Papatheodorou, T. S. (2005). A Methodology for Conducting Knowledge Discovery on the Semantic Web. In *Proc. of 16th ACM Conference on Hypertext and Hypermedia (Hypertext 2005), International Workshop on Adaptive and Personalized Semantic Web*. Salzburg, Austria.

Koutsomitropoulos, D. A., Paloukis, G. E., & Papatheodorou, T. S. (2007). Ontology-based Knowledge Acquisition through Semantic Profiling. An Application to the Cultural Heritage Domain. In *Proc. of the 2nd International Conference on Metadata and Semantics Research (MTSR 2007)*. Corfu, Greece. CD-ROM.

Lassila, O., & Hendler, J. (2007). Embracing "Web 3.0". *IEEE Internet Computing* 11(3), 90-93.

Loosley, C. (2006). *Rich Internet Applications: Design, Measurement, and Management Challenges*. White Paper. Keynote Systems.

Luke, S., Spector, L., & Rager, D. (1996). Ontology-Based Knowledge Discovery on the World-Wide Web. In A. Franz, & H. Kitano (Eds.), *Working Notes of the Workshop on Internet-Based Information Systems at the 13th National Conference on Artificial Intelligence (AAAI '96)* (pp. 96-102). AAAI Press.

Manola, F., & Miller, E. (2004). Resource Description Framework (RDF) primer. *W3C Recommendation*. Retrieved 10 February 2004, from http://www.w3.org/TR/rdf-primer/.

Mika, P. (2005). Ontologies are us: A unified model of social networks and semantics. In *Proc. 4th International Semantic Web Conferences (ISWC05)* (pp. 522–536). Galway, Ireland.

O'Reilly, T. (2005). *What is Web 2.0 – Design patterns and business models for the next generation of software*. Retrieved 30 September 2005, from http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html.

Oren, E., Delbru, R., & Decker, S. (2006). Extending faceted navigation for rdf data. In I. Cruz and S. Decker (Ed.), *Proc. 5th International Semantic Web Conference (ISWC06)* (pp 559–572). Athens, Greece.

Smith, M. K., Welty, C., & McGuinness, D. (2004). OWL Web Ontology Language Guide. *W3C Recommendation.* Retrieved 10 February 2004, from http://www.w3.org/TR/owl-guide/.

Tenenbaum, J. M. (2006). AI Meets Web 2.0: Building the Web of Tomorrow, Today. *AI Magazine* 27(4), 47-68.

Tziviskou, C., & Brambilla, M. (2007). Semantic personalization of web portal contents. In *WWW '07: Proceedings of the 16th International Conference on World Wide Web* (pp. 1245-1246). New York, USA. ACM Press.

## KEY TERMS & DEFINITIONS

**Web 3.0:**
Web 3.0 is a term used to describe the future of the World Wide Web. Following the introduction of the phrase "Web 2.0" as a description of the recent evolution of the Web, many technologists, journalists, and industry leaders have used the term "Web 3.0" to hypothesize about a future wave of Internet innovation.

**3-tier Architecture:**
3-tier architecture is a client-server architecture in which the user interface, functional process logic ("business rules"), computer data storage and data access are developed and maintained as independent modules, most often on separate platforms.

**Semantic Web:**
The Semantic Web is an evolving extension of the World Wide Web in which the semantics of information and services on the web is defined, making it possible for the web to understand and satisfy the requests of people and machines to use the web content. It derives from W3C director Tim Berners-Lee's vision of the Web as a universal medium for data, information, and knowledge exchange.

**Web 2.0:**
Web 2.0 is a term describing the trend in the use of World Wide Web technology and web design that aims to enhance creativity, information sharing, and, most notably, collaboration among users.

**Ontology:**
An ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts. It is used to reason about the properties of that domain, and may be used to define the domain. Ontologies are used as a form of knowledge representation about the world or some part of it.

**Knowledge system:**
A knowledge system (a.k.a. knowledge-based system) is a program for extending and/or querying a knowledge base. A knowledge base is a collection of knowledge expressed using some formal knowledge representation language.

**Mash-up:**
A mash-up is a web application that combines data from more than one source into a single integrated tool.