Fly-By-OWL: A Framework for Ontology Driven E-commerce Websites

Azhar Jassal, David Bell

Fluidity Research Group
Brunel University, Uxbridge, West London, UB8 3PH, UK
{Azhar.Jassal, David.Bell}@brunel.ac.uk

Abstract: Fly-By-OWL allows Web developers to create "ontology-driven" E-commerce websites that can harness live reasoning. It aims to make the Semantic Web's underlying technologies (ontologies and reasoning) relevant to Web developers. To demonstrate Fly-By-OWL, the "Semantic Pizza Store" is presented, an example store that uses the output of the pizza ontology tutorial as its knowledge base. By making use of inferences, products can be categorised dynamically and product customisation can rely upon consistency rather than hard-coded rules.

Keywords: Ontology, Semantic Web, OWL, Web Application, CGI, Dynamic Website, E-commerce, LAMP, Database, DBMS, Pizza

1. Introduction

The World Wide Web was the "killer app" that spread the internet to every home and office across the world. In the beginning, a website was just a collection of interlinking pages written in Hypertext Mark-up Language (HTML). The introduction of the Common Gateway Interface (CGI) specification in 1993 allowed a website to become both dynamic and interactive. From then on the design of a Web page could be separated from its content. Pages were no longer static; they were generated on-the-fly with content that was customised for each user's request. This spurred the advent of a whole range of Web applications, allowing everything from Web-based email (the original HoTMaiL) and e-commerce sites (Amazon) to become possible. While many technologies exist today that allow programming for the Web, from servlets to dynamic scripting languages, it was CGI that long ago led the way. With the design of a website separated from its content, relational databases found their place as the default backend of Web applications.

The extensive features provided by a fully fledged database management system (DBMS) made them the Web developer's default choice for not just data but all content handling. From the advent of the dynamic website to the present day, databases are used to hold everything needed to feed a Web application, from user details and visitor information, to content such as product information for an e-commerce store and the messages that somebody posts on their Blog. It is databases

that hold the content that feeds today's Web. Yet, the Web itself has held limits that stem from its original design, by being based upon Hypertext Mark-up Language (HTML), nothing is said about what the data is for, i.e. about its semantics [1]. The biggest problem with the Web is that "information is dumb; the data contained in websites does not know what it is" [2]. To address this, the Semantic Web project was initiated in 2001 and aimed to bring structure to the meaningful content of Web pages [3].

The paper begins by analysing if Semantic Web technologies have found any utilisation within the backend of a present day Web application. Related work is discussed, noting the position of ontologies within those Web applications and their significance to its overall operation. Fly-By-OWL is then introduced; a framework that allows developers to create "ontology-driven" Web applications that can harness the potential of live reasoning. The first implementation of Fly-By-OWL focuses on an e-commerce context, by specifically allowing developers to create "ontology-driven" business-to-consumer (B2C) e-commerce websites. The e-commerce context is chosen to "emphasise upon practical application" which is necessary to work towards the widespread adoption of the Semantic Web [4]. The framework is then demonstrated with the "Semantic Pizza Store", which uses the ontology output from the Protégé pizza tutorial [5].

2. Finding a place for Semantics in Today's Web

It has been almost 20 years since the initial advent of the Web. According to the Netcraft Web Server Survey (July 2009), there are now 240 million hostnames running a publicly accessible Web server, compared to 24 million in October 2000 [6]. With 8 years passed since the commencement of the Semantic Web project, how many mainstream Web applications are now employing Semantic technologies? And where have developers, those responsible for serving relevance to real users, positioned them within their Web applications? O'Reily [7] wrote that "the Internet is the most striking example of a marketplace, full of economic opportunity, which has grown out of the open-source software community". As of July 2009, Apache, an open source Web server, holds the dominant 47% share of the market [6] and forms part of the larger LAMP stack for Web applications. LAMP is a free open-source software bundle which provides the principal components used to build a viable general purpose web server. Once mostly used for small-scale Web development, it has advanced its way into mainstream corporate software development, being used by Google and Yahoo to build search applications, while Lufthansa and the Sabre Travel Network used it to develop travel reservations systems [8].

In order to gather a snapshot of how developers are using Semantic technologies, six popular LAMP based projects with highly active user communities are analysed in Table 1, to determine whether any features of these packages utilise ontologies.

Support is categorised as either: standard (official feature), planned in roadmap (upcoming standard feature) or unofficial add-on (community maintained contribution).

Table 1. Ontology utilisation within six widely used LAMP based Web applications

Project	Description	Standard	Roadmap	Add-on
Drupal	Content Management System (CMS) used as a back-end for websites. Is followed by over 550,000 registered users and is used by thousands of websites. User contributed add-on: Drupal SIOC			✓
Joomla	Another CMS, used as a back-end to websites. Like Drupal, used by many prominent websites, followed by over 300,000 registered users. User contributed add-on: GoodRelation's for VirtueMart (a Joomla e-commerce extension)			✓
phpBB	Most widely used open-source bulletin board system in the world, followed by over 350,000 registered users. User contributed add-on: phpBB SIOC			✓
osCommerce	Open source out-of-the-box e-commerce solution with a community of 210,000 registered users. User contributed add-on: GoodRelation's			✓
vBulletin	Commercial bulletin board software used by thousands of websites, followed by a community of 200,000 registered users. User contributed add-on: vBulletin SIOC			✓
WordPress	Open source blog publishing application and CMS used by millions of websites. User contributed add-on: WordPress SIOC			✓

In an effort to find answers to the two questions posed beforehand, these six popular LAMP based Web applications were examined. The results show a clear picture, while all of the applications have a community contributed add-on that utilises ontologies in one form or another; the actual developers of the applications have no plans to use ontologies within a standard feature. This makes a simple fact apparent: while the Semantic Web does have an active research community that wishes to captivate the use of it's technologies within these projects, the technologies themselves have not proved their present real world benefit to developers. McBride [4] wrote that in order to step towards widespread adoption of Semantic Web technologies, we must "emphasise practical applications" and that we must start to "develop applications now". That was 7 years ago and while the Web continues to show unprecedented growth, the same unfortunately cannot be said about the Semantic Web. It has still not made any impact that can be felt by a real world Web user, i.e. common man.

What these six popular Web applications share in common is that they are all "database-driven". The LAMP stack itself positions the database as a fundamental building block of a Web application, with the "M" in LAMP standing for MySQL, an open source DBMS. The initial advent of dynamic websites, where design could be separated from content, was a giant leap. With it, the DBMS became the default "content-handler", and so it seems what was made default with that great leap, has continued to remain default into the present day. The Web, above any other development platforms before it, is led by example. This can be blamed to being a by-product of its rapid growth. In 2004, the World Wide Web Consortium (W3C) selected a standard ontology language, OWL (Web Ontology Language). It is based upon Description Logic (DL) and exploits existing work on languages such as OIL and DAML+OIL [9]. With an ontology language chosen, it would have been expected for some Web applications to begin to adopt ontologies as their knowledge bases over a DBMS, but ontologies have still not become the backend to any noticeable real world Web application.

With Semantic technologies still not being accommodated for, is the Web of today any different from when CGI first made its advent? Perhaps the Web has been stuck in an endless loop; by using in essence the same building blocks from the advent of CGI that cannot accommodate semantics. Rob McCool, the author of "httpd" which later became Apache wrote that "without radical simplification, the Semantic Web will continue to see limited participation and few compelling applications" [10]. In order to move the field forward, this paper proposes an approach for creating "ontology driven" e-commerce websites that goes back to basics. It focuses upon being directly relevant to a general purpose Web developer. It allows novice Web developers to harness ontologies as their primary knowledge base, rather than a DBMS and make use of live reasoning. By allowing developers to "begin developing now" and by "emphasising practical applications" [4], it aims to progress adoption of the Semantic Web by making its underlying technologies more relevant to Web developers, as they build the applications that are relevant to real Web users.

3. Ontology utilisation within Web Applications

As can be seen in Table 1, the Semantic Web community has been active in contributing add-ons for popular Web applications that enable them to make some utilisation of Semantic technologies. An add-on or extension is an optional component, and installed by a user to add specific functionality that the developers did not deem significant enough to have to offer as standard. The Semantically-Interlinked Communities (SIOC) Initiative uses an ontology to represent "social data" in RDF [11]. By offering add-on's for four of the six Web applications listed in Table 1, large ontologies would be generated by exporting content. By offering its add-on "exporters" for various applications, SIOC envisions being able to interlink these "independent" and separated communities. The ontologies themselves do not play any role vital to the function of the overall Web applications, and serve a different purpose to the scope of the "ontology-driven" progression being proposed by this paper.

The remaining two Web applications listed in Table 1 make their ontology utilisation with GoodRelation's, a lightweight ontology for annotating e-commerce offerings [12]. GoodRelation's provides a vocabulary for describing the types of goods and the terms and conditions of items and services offered on the Web. It is an accepted vocabulary of Yahoo! SearchMonkey, to accommodate structured data for their search engine. The add-on's for the e-commerce Web applications (an extension in the case of Joomla) generate structured data following the GoodRelation's vocabulary from the product data already present within the Web applications back-end database. GoodRelation's provides a suitable vocabulary for the knowledge base of an "ontology-driven" e-commerce Web application. This paper focuses on a framework that allows the creation of such websites, where the Web developer is free to choose the most appropriate vocabulary for their knowledge base. While SOIC and GoodRelation's were projects that happened to contribute add-on's, there are also other projects that concern using ontologies within Web applications.

Stojanovic et al. [13] provides a reverse engineering approach to migrating data-intensive websites to the Semantic Web. By transforming a present day back-end relational database model into corresponding ontological structures, content is mapped from the database into an ontology. OntoWeaver, a website design framework, uses ontologies to drive the design and development of data-intensive websites [14]. OntoWebber is a model-driven ontology-based system architecture for creating data intensive Web sites and portals [15]. Of these projects, the most closely related to the scope of this paper is OntoWiki, an open source semantic wiki Web application which facilitates the visual representation of a knowledge base as information maps [16]. Within OntoWiki, Erfurt is being developed, a Semantic Web toolkit with a native reasoner (and DIG capabilities) for the PHP programming language. While the context is wiki's, the usage of ontologies as the primary knowledge base that drives the Web application makes it the most relevant to this paper. The simultaneous creation of a Semantic toolkit for PHP, which along with Perl and Python makes the "P" in LAMP, could be another vital element in enabling Semantic technologies to find a more prominent position within Web applications.

Bearing in mind the call for "rapid simplification" or the Semantic Web having to face limited partition [17], we present a framework that goes back to basics and focuses on being relevant to even casual Web developers. By bringing together the standard ontology language (OWL), semantic reasoners (e.g. FaCT++, Pellet) and ontology editors (e.g. Protégé), the Fly-By-OWL framework allows the most novice Web developer to create an "ontology-driven" e-commerce website. It allows Web developers to harness the artificial intelligence (AI) capabilities provided by reasoning in their websites. It introduces the "ontology-driven" concept in an e-commerce context to "emphasise upon practical application" [4]. With Fly-By-OWL, developers can manipulate a knowledge base (with live reasoning on-the-fly) just as their applications would have previously interacted with a DBMS. Fly-By-OWL aims to make the creation of "ontology-driven" websites a possibility for even casual Web developers and practically demonstrate the leap provided by using ontologies over present day means. To the best of our knowledge, the implementation will be the first

to allow an everyday Web user to interact with an OWL knowledge base with live reasoning within some context, e.g. e-commerce. Allowing Web developers to harness the abilities of a reasoner and progressing from "database-driven" to "ontology-driven" will allow us to step closer to the Semantic Web.

4. Fly-By-OWL: Ontology Driven Websites

The Fly-By-OWL framework allows even a novice Web developer to create "ontology-driven" Web applications. The initial implementation of the framework focuses on using it to create e-commerce stores. With product information presently held in a database, a Web developer would write a catalogue/ shopping cart which would interact with a DBMS. Fly-By-OWL presents the Web developer with a data-model of an OWL knowledge base (with inferences) which can then be presented through HTML however the developer envisions. The knowledge base can be queried using Manchester Syntax [18]. Using ontologies allows limitations imposed by the present day "database-driven" approach to be overcome. For example, product's can be categorised on-the-fly using equivalent classes in the ontology, by using reasoning and inferences which place products within appropriate categories. Product specifications can be customised without any hard coded rules as reasoning can indicate whether a class is now inconsistent.

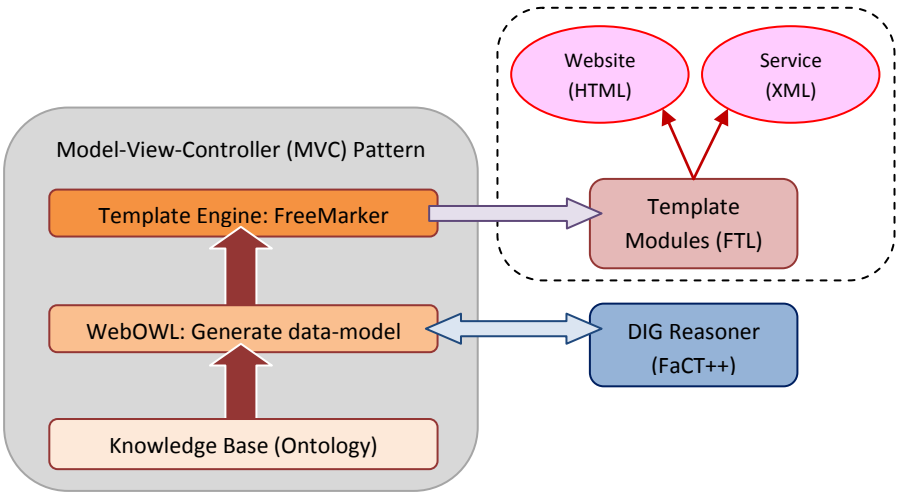

Figure 1: Fly-By-OWL Framework

The framework is compromised of three layers (see Fig. 1) and uses a Model-View-Controller (MVC) pattern to isolate "content", in essence business logic (the knowledge base) from "design". While the framework is able to produce output in

either HTML (Web browser, for humans) or XML (Web services, for machines), the first implementation focuses on HTML, to "emphasise upon practical application" [4] within the context of e-commerce websites. Fly-By-OWL requires a live constant connection (over DIG) to a reasoner, which runs as the backend reasoning server.

4.1 Bottom Layer: Knowledge Base (Ontology)

The bottom layer of the framework compromises of the ontology itself. The knowledge base can be created in Protégé, an open source ontology editor. The implementation of the framework supports both OWL and OWL 2 ontologies. In a present day Web application, the knowledge base could be served by anything from a flat file up to a DBMS. By using an ontology, artificial intelligence capabilities can be harnessed by the Web developer that with semantics, are more rich in scope and ability when compared to present day means.

4.2 Middle Layer: Data-model Generator

The middle layer of the framework interacts with the knowledge base and a reasoner (over DIG) to generate a data-model. Whereas the behind-the-scenes of a current day dynamic website may use a DBMS as a database server, Fly-By-OWL uses FaCT++ as a reasoning server. Once inferences have been made, the data-model is generated following a set specification. A custom query in Manchester Syntax can be passed into this layer. If one is received, it is made equivalent to a temporary class which is added to the knowledge base that holds results after reasoning.

4.3 Top Layer: Template Engine

The top layer of the framework uses FreeMarker [19] as the template engine and is able to produce output for either a website (HTML) or Web service (XML). The output is customised by scripting templates in either HTML or XML with the FreeMarker Template Language (FTL). FTL allows the user to manipulate and fetch elements from the data-model. The Web developer can place these elements however they envision through the use of templates. Template "modules" are created that each present both different information and functionality to a user.

4.4 Template modules

Template modules allow a Web developer to create a Web application with Fly-By-OWL just as they would with any other dynamic scripting language (e.g. PHP, ASP, and JSP). For example, the template module "index" may contain an initial welcome page to the website, whereas the template module "products" may list items and the module "customise" may provide the functionality to enable a user to customise a products properties (with reasoning used to check for consistency). Using modules makes creating a website with Fly-By-OWL no different from how Web applications have always been created with dynamic scripting languages. Pages hold individual functionality, and once interlinked create a complete Web application. Some

examples of these individual pages can be: “home” (home page), “catalogue” (product catalogue), “cart” (visitors shopping cart), “checkout” (start of checkout), “complete” (end of checkout).

4.5 Fly-By-OWL in practice

To illustrate how Fly-By-OWL works in practice, Fig. 2 displays the inferred class hierarchy of the ontology output from the Protégé pizza tutorial [5]. Here, the “NonVegetarianPizza” class shows its inferred results; pizzas that contain a meat or seafood topping. From an e-business perspective, a Web developer may want to use the inferred results of this class as a non-vegetarian category on a pizza ordering e-commerce website.

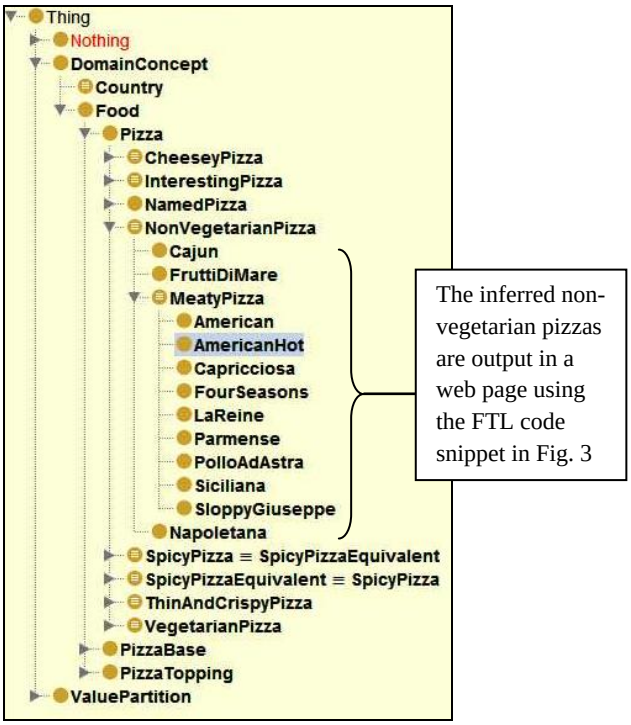

Figure 2: The pizza ontology tutorial (inferred class hierarchy)

By using FTL, “NonVegetarianPizza” can be found in the data model. Iterating through its contents will allow a Web developer to display the pizzas that contain a meat or seafood through a “dynamic category”. The category is dynamic as the pizzas within it have not been specified by humans, but are placed there due to their properties and through inference. An excerpt from a template which makes use of the inferred results of the “NonVegetarianPizza” class, written in HTML with FTL scripting can be seen in Fig. 3. This is a code snippet from the “pizzas” page of the “Semantic Pizza Store” (discussed further in section 1.4). It demonstrates how to

output the inferred subclasses of “NonVegetarianPizza”. Following the data-model specification, elements can be retrieved from the knowledge base and output to the user. The code snippet in Fig. 3 iterates through the data-model at “NonVegetarianPizza”, displaying a box for each pizza, made from a table that contains its name and toppings (inferred pure classes). FTL provides all of the common calls a Web programmer would require, such as ‘if’ and ‘else’ clauses, ‘for’ and ‘while’ loops, among many other functions that allow the manipulation of the data-model.

```
<#set whichClass = NonVegetarianPizza>
<#list whichClass?keys as pizzaName>
<table>
<tr>
<td><b>${pizzaName}</b></td>
</tr>
<tr><td>
        <table>
        <tr><td>

        <!-- Toppings //-->
        <table>
         <#list whichClass[pizzaName].PureClasses?keys as topping>
                <tr>
                <td>${topping}</td>
                </tr>
        </#list>
        </table>
        <!-- End Toppings //-->

        </td></tr>
        </table>
</td></tr>
</table>
</#list>
```

Figure 3: Snippet of the Pizza template module (FTL in red)

The first implementation of the Fly-By-OWL framework focuses on using it to create ontology driven business-to-consumer (B2C) e-commerce stores. This context was chosen as it can demonstrate in practice the benefits gained from being “ontology-driven”. Developers can explore the advanced product handling abilities gained by being able to reason and make inferences, and not have to rely on hard coded rules. The stores created using Fly-By-OWL will be the Web’s first ontology driven websites in real world context that require live reasoning to produce every page of output. By placing ontologies in real world Web context, the framework aims to make them relevant to Web developers. Once a product ontology has been prepared, for instance in Protégé, a Web developer must then create appropriate templates that showcase the products and allow the user to undertake common e-commerce store functions (such as browsing products, viewing additional product details). The implementation of the framework includes functionality within it that provides common features expected in an off-the-shelf e-commerce platform, such as a shopping cart, which is accommodated within the specification of the data-model.

5. Semantic Pizza Store

The example store that has been created to demonstrate the framework and its capabilities is the "Semantic Pizza Store", based upon the output of the well known pizza ontology tutorial, written in Manchester by Matthew Horridge. The tutorial teaches ontological concepts to new Protégé users [5]. The ontology output from following the tutorial is used as the knowledge base for the store. The pizzas are offered for sale, with live inferences being made to categorise pizzas as Vegetarian, Non-Vegetarian, Spicy, etc and allowing the user to customise standard pizzas and create their own. The knowledge base can be queried on the fly in "Manchester Syntax", for example requesting all pizza's from a specific country or of a certain spiciness. The queries can be either input by the user and POST, or they can be passed in via an encoded URL in a GET request. The "Semantic Pizza Store" demonstrates both how ontologies are appropriate as the backend knowledge base to sell the products in question, pizzas and how they overcome the limitations of a product database. Fig. 4 shows a screen capture of the home page of the "Semantic Pizza Store".

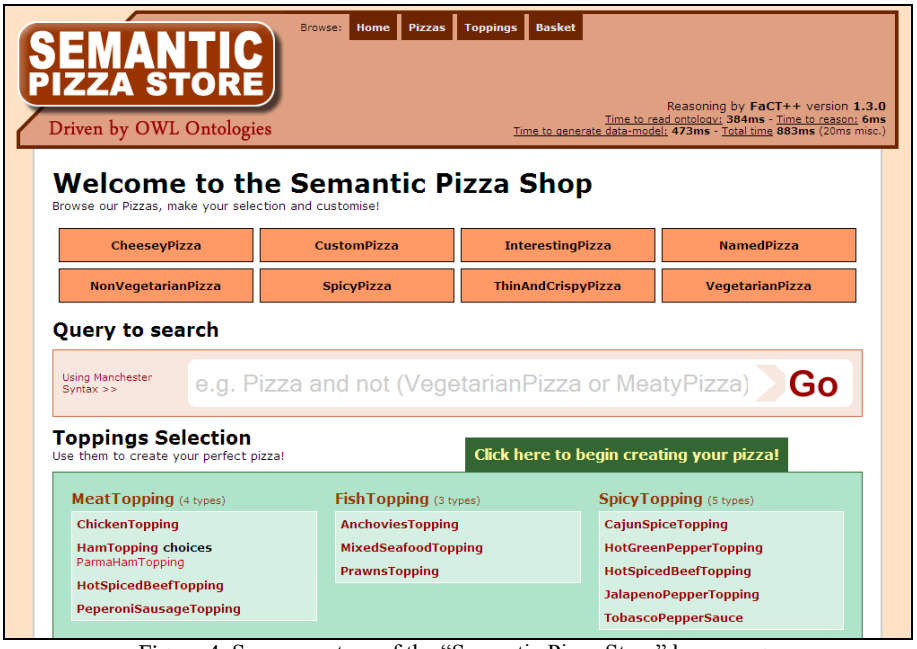

Figure 4: Screen capture of the "Semantic Pizza Store" home page

The popularity of the pizza ontology tutorial makes it appropriate to use in demonstrating Fly-By-OWL. Its concepts will be familiar to most researchers in the field. Fig. 4 shows the home page module ("index") of the "Semantic Pizza Store". The pizza subclasses are fetched from the knowledge base and displayed as clickable boxes, acting as "dynamic categories", allowing the user to browse pizza's of those types. The results of these classes are inferred. A user can query the knowledge base

in Manchester Syntax, and toppings are displayed so a user can begin to create their own pizza. Additional functionalities of the "Semantic Pizza Store" include customising the preset pizzas, creating half and half's and adding pizzas to a shopping cart from where the user can proceed to a checkout, among other functionality expected in a typical online pizza ordering service. The template modules of the example store are all written in HTML and make use of CSS and JavaScript, similar to other current day websites. The data-model is manipulated through FTL scripting, as was shown in the code snippet in Fig. 3.

The source code of the templates that are used by the "Semantic Pizza Store" is viewable online at the Fly-By-OWL website. Fig. 5 shows the pizza module, displaying the subclasses of "NonVegetarianPizza" (results inferred). It is using the code snippet from Fig. 3 to generate the pizza description boxes. The objective of the example store is to present the capabilities of the framework to Web developers. The framework itself can handle any OWL or OWL 2 ontology that is loaded as the knowledge base. The framework does not treat the pizza ontology different to any other knowledge base. To create a store using the Fly-By-OWL framework, a Web developer must upload their ontology and then create appropriate template modules using the data-model specifications and FTL scripting. This allows a Web developer to present the concepts within the knowledge base however they best envision.

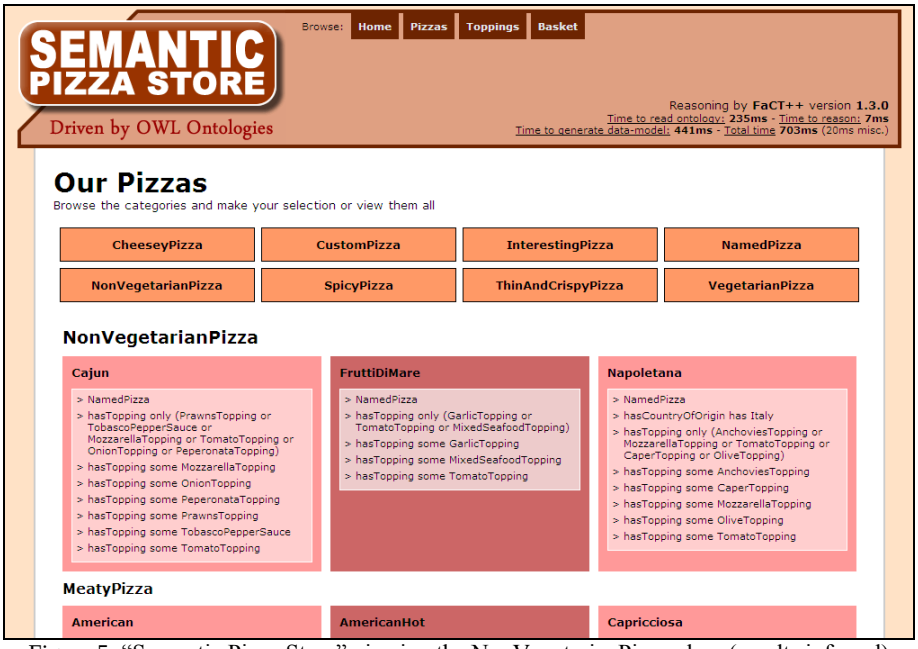

Figure 5: "Semantic Pizza Store" viewing the NonVegetarianPizza class (results inferred)

Fly-By-OWL allows the creation of "ontology-driven" e-commerce websites while emphasising its practical relevance to web developers. During development, the pages output to a user were all generated in under one second. The lengthiest operations

within a page load are reading the ontology itself and the generation of the data model. While Web application platforms are generally tried-and-tested, the "ontology-driven" with live reasoning concept is new and further research is required to understand how it will cope under various loads, and the hardware/ software setup required to best handle such traffic. Performance monitoring a Fly-By-OWL store under various traffic loads has been identified as a further research topic. The size of the ontology being used as the knowledge base and it's affect on load times will also be studied. With further development of the platform's architecture, load times are intended to be comparable to any database-driven Web application.

6. Research agenda

With the first implementation of Fly-By-OWL focused towards an e-commerce context, research questions arise in regards to how the "ontology-driven" concept will cope against traditional tried-and-tested back-ends (e.g. a DBMS). A number of research questions arise in regards to its real-world usage and further development.

- How are page load times affected by traffic?
- How are page load times affected by size of the knowledge base ontology?
- What types of ontology will be used as the knowledge base? E.g. GoodRelation's vocabulary and how will they be handled by Fly-By-OWL?
- How should the platform be expanded beyond the e-commerce context and become more applicable for general Web applications?
- What kind of Web services can be created by using XML templates with FTL scripting and how should they harness reasoning?

7. Conclusion

This paper has presented Fly-By-OWL, a framework that enables the creation of "ontology-driven" Web applications, with its first implementation aimed specifically towards e-commerce stores. By following calls for "rapid simplification" of the Semantic Web, this paper aims to make ontologies relevant to Web developers. By starting with an analysis of ontology usage within six popular LAMP based Web applications, it became apparent that the developers of those applications found no place for ontologies within any standard or road mapped feature. While the Semantic Web community was active in creating add-on's for those applications that allowed them to make use of ontologies in some form, the scope of their depth was trivial and they did not play any role vital to the operation of the applications. With Fly-By-OWL, e-commerce store's can be created for the present day Web that are driven by an ontology. With live reasoning, present day limitations experienced when using a database can be overcome. The paper demonstrated some features such as using inferences to dynamically categorise products and customising products without hard coded rules but verifying consistency. The paper also showcased the "Semantic Pizza

Store", an example e-commerce store based upon the pizza ontology tutorial created to demonstrate the framework. A research agenda has been formulated that looks to address some of the initial questions posed by the "ontology-driven" concept and how to further Fly-By-OWL's overall relevance and scope.

8. Project on the Web

The Fly-By-OWL project website and the "Semantic Pizza Store" are hosted at Brunel University, U.K. and can be found online at http://www.flybyowl.org. You are welcome to use the framework to create your own "ontology-driven" e-commerce stores and interact with our online community.

9. References

[1] P. Warren, The next steps for the web: Putting meaning into the web. *Computing & Control Engineering 1(2),* pp. 27-35, 2003.

[2] D. Bradbury, Unweaving the tangled web of dumb data. *Computer Weekly 10*pp. 1-7, 2003.

[3] T. Berners-Lee, J. Hendler and O. Lassila, "The Semantic Web," *Sci. Am.*, May. 2001.

[4] B. McBride, "Four steps towards the widespread adoption of a semantic web," in *ISWC '02: Proceedings of the First International Semantic Web Conference on the Semantic Web,* 2002, pp. 419-422.

[5] M. Horridge, H. Knublauch, A. Rector, R. Stevens and C. Wroe, "A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plug-in and CO-ODE Tools Edition 1.0," August. 2004.

[6] Netcraft. (2009, 07/28/2009). July 2009 Web Server Survey. *2009(08/10),* Available: http://news.netcraft.com/archives/2009/07/28/july_2009_web_server_survey.html

[7] T. O'Reilly, "Lessons from open-source software development," *Commun ACM*, vol. 42, pp. 32-37, 1999.

[8] G. Lawton, "LAMP lights enterprise development efforts," *Computer*, vol. 38, pp. 18-20, Sept. 2005.

[9] I. Horrocks, "Semantic web: The story so far," in *W4A '07: Proceedings of the 2007 International Cross-Disciplinary Conference on Web Accessibility (W4A)*, 2007, pp. 120-125.

[10] R. McCool, "Rethinking the Semantic Web, Part 2," *IEEE Internet Comput.*, vol. 10, pp. 96-95, 2006.

[11] J. G. Breslin, S. Decker, A. Harth and U. Bojars, "SIOC: an approach to connect web based communities," *Int.J.Web Based Communities*, vol. 2, pp. 133-142, 2006.

[12] M. Hepp, "GoodRelations: An ontology for describing products and services offers on the web," in *EKAW '08: Proceedings of the 16th International Conference on Knowledge Engineering*, 2008, pp. 329-346.

[13] L. Stojanovic, N. Stojanovic and R. Volz, "Migrating data-intensive web sites into the semantic web," in *SAC '02: Proceedings of the 2002 ACM Symposium on Applied Computing*, 2002, pp. 1100-1107.

[14] Y. Lei, E. Motta and J. Domingue, "OntoWeaver: An ontology-based approach to the design of data-intensive web sites," in *Hypertext/Hypermedia Handbook*, 2005, pp. 244-262.

[15] Y. Jin, S. Decker and G. Wiederhold, "OntoWebber: Model-driven ontology-based web site management," in *Semantic Web Working Symposium (SWWS)*, 2001.

[16] M. Hepp, D. Bachlechner and K. Siorpaes, "OntoWiki: Community-driven ontology engineering and ontology usage based on wikis," in *WikiSym '06: Proceedings of the 2006 International Symposium on Wikis*, 2006, pp. 143-144.

[17] R. McCool, "Rethinking the Semantic Web, Part 1," *IEEE Internet Comput.*, vol. 9, pp. 88-87, 2005.

[18] M. Horridge, N. Drummond, J. Goodwin, A. L. Rector, R. Stevens and H. Wang, "The Manchester OWL Syntax," in *OWLED; CEUR Workshop Proceedings*, 2006.

[19] FreeMarker Project, "FreeMarker: Java Template Engine Library," 2009(08/10), Available: http://www.freemarker.org