Modeling a Complete Ontology for Adaptive Web Based Systems Using a Top-Down Five Layer Framework

Bujar Raufi, Florie Ismaili, Xhemal Zenuni
South East European University, Ilindenska 1200 Tetovo, Macedonia
E-mail(s): b.raufi@seeu.edu.mk, f.ismaili@seeu.edu.mk, xh.zenuni@seeu.edu.mk

Abstract. The web pages of today strive to think out of the box by creating different look-and-feel for each particular user. Most of them still are a pure cast of HTML tags, however some web applications tend to organize their structure and content based on user access patterns. These systems are known as Adaptive web based systems.

In this paper we present a complete ontology (Concept Domain) for Adaptive Web Based Systems based on Top-Down five layer framework. We also test the inference in such ontology and conclude that the findings fulfill the requirements of our proposed framework and that this ontology is suitable for general purpose Adaptive Web-Based Systems.

Keywords. Adaptive Web-Based Systems, Ontology, Semantic Web

1. Introduction

Most of the web pages we see today represent a collection of HTML tags organized in "one-size-fits-all"[3] maner. Different users however have various preferences even for the same web site seen through longer period of time. In order to meet the above mentioned requirements, a system that rearranges its site's internal structure and content needs to be created. These types of web systems are said to be adaptive.

Adaptive Web-Based Systems can be generalized into three main categories summarized as:
- Adaptable Web Based Systems that arrange their content or structure based on preliminary feedback from the user. This feedback is ensured in various ways like logging in or filling up web forms. These systems and approaches used are also called proactive systems or approaches [12].
- Adaptive Web-Based Systems on the other hand tend to arrange their content or structure without direct "invasion" from the user. The main source used to extract user behaviour is the web server logs. These systems or approaches are known also as reactive systems or approaches [12].
- And finally, in Dynamic Web Based Systems, where besides them being adaptive, there is no predefined data presentation, but the adaptation and page synthesis is being performed out of elementary pieces of information (atomic units) such as: Text, Multimedia, Links etc.

The last type of adaptive web based system is of particular importance to our research as we are going to see later when we present our framework and ontology. The contribution of this ontology is that it offers expressivity, modularity and interoperability as required by our framework; as well as its suitability for general-purpose Adaptive Web-Based Systems.

The rest of this paper is organized as follows:
In section 2, we give a brief description of related work on semantic web ontologies regarding adaptive web based systems with short description of our five layer top-down framework. In section 3, a complete ontology for adaptive web based systems that comply with our framework is presented. In section 4 some rules and inference testing is being performed and section 5 concludes the paper.

2. Related work

Most of the related work done concerning Adaptive Web-Based Systems is on the design approaches. Some of the approaches can be abstracted as:
- Complete data-centric approach where the main focus is the data, its organization and relationship. This approach is mostly used in developing a model-driven context aware web applications as described in [5].
- The second approach is a web mining approach where data is exploited and manipulated with no clear emphasis on
knowledge or in the best case, the knowledge discovered needs additional explanation from a domain expert [6,10].

Finally the third approach is completely knowledge-centric, where data is abstracted and generalized in higher levels of application logic. This approach is exemplified in [15,4,7].

The overhead approaches suffer from several drawbacks due to one way treatment of the problem i.e. they are either completely data centric or completely knowledge centric. In [11] we presented a top-down five layer framework that offers flexibility, expressiveness, interoperability and modularity to address the above mentioned problems. It is coined as top-down because it starts with the data layer at the top and narrows down to the adapted content by taking into consideration other layers that lay in between. The layers are considered in such a way to take into account both data and knowledge (concepts about data). This framework consists of five layers as depicted below in Fig. 1.

![Figure 1. A Top-Down Five Layer Framework For Adaptive Web Based Systems](image)

The first layer called a data layer is consisted of loosely coupled atomic data units. The units can be texts, multimedia files (audio, video or animations) or any other describable content. These units represent the building blocks of a web page. The second layer represents a layer of concepts and concept relationships gathered from atomic units. The ontology that we present in this paper is designed to function in this layer. The third layer takes care of the information gathered from users that accesses the page. The information acquired here reside on the user information repository, which stores user access on the page's content in the form of web logs, as well as pattern repository that uses data mining techniques for extracting and accumulating user access behavior. The adaptation layer consists of set of rules on how to perform adaptation and finally presentation layer rearranges the web site's content or structure based on the rules from adaptation layer.

Designing a complete ontology for adaptive web based systems is not a trivial task. Most of the ontology that exist today focuses mostly on modeling the content and relationship of content on such systems. For example an ontology for the content of adaptive web based systems has been presented in [2]. Its ontology consists of Domain Model and a Domain Dependent User Model. However this model uses a deprecated user domain, considering that it does not take into account the user access behavior, but only the user view related directly to concept in the domain model. As we will see in the next section our proposed ontology completely incorporates the user as a consisting part of adaptation.

Another ontology approach is presented in Adaptive Personal Information Environment (a-PIE) which is based on Fundamental Open Hypermedia Model (FOHM) [9]. This approach is focused more on enabling users to search information based on ontologically defined concepts, rather than creating an ontology that will completely express an adaptive web based system.

In the next section will give a complete ontology that will fulfill the requirements of the framework given earlier.

3. Modeling a complete ontology for Adaptive Web Based Systems

The main idea behind this ontology is to envision a web site that will be consisted of loosely coupled atomic data units. Based on user preferences, these atomic units can be gathered to create more composite unit (pages) that will be presented in an adapted way. The ontology consists of concept domain and user domain. In concept domain we store atomic units as concepts as well as concept relationships. These concepts have their own information weights (ratings) and can be aggregated into pages. The
A generalized model of the ontology is illustrated as in Fig. 2.

The user domain is consisted of concepts concerning the user like: user requests, sessions and user preferences.

![Figure 2. The Generalized Model of the Complete Ontology](image)

The steps involved in creating the ontology for the above mentioned method encloses the following:

1. Definition of concept classes and hierarchies.
2. Definition of concept properties
3. Specifying the relationship between concepts
4. Definition of concept instances for some classes of the ontology

The ontology have been defined using OWL DL\(^1\) Language, which allows enough expressiveness as well as inference possibilities. The definition of concept classes, relationships and instances is done using Protégé ontology editor and knowledge base system[14].

### 3.1. Definition of Concept Classes and Hierarchies

The complete ontology consists of core class and inference class. The core class consists of five main classes namely: AtomicUnit, Page, Preference, Rating and User. The inference class called Visits will be used for testing the reasoning in the core classes. This will be explained in section 4.

The AtomicUnit class can have a type, which represents the general type of data that the atomic unit can be consisted of, like: image, multimedia (audio, video or animation) and text. To completely describe the atomic unit ontology additional classes are needed as well. For example the Position subclass that describes the position of atomic unit within a page and Link subclass that creates the relationships between these concepts.

Other classes in the core involves the Page class which can be composed from several atomic units, the Preference class consisted of several sub classes namely general preferences that the users might seek in a page, and Rating class that gives the atomic unit an information weight like: High, Medium or Low importance. The complete class hierarchy as shown from Protégé is presented in Fig. 3.

![Figure 3. The Asserted model of the Ontology](image)

\(^1\)OWL DL – (DL – Description Logic)Represents a sub language of Web Ontology Language(OWL) that allows definition of logic rules and efficient reasoning support.
3.2. Definition of Concept Properties and Relationships

The concept or class properties relates individuals from one class (domain) to another class (range) creating this way a relationships between concept classes. Without class properties, the inference in the ontology would not be possible, considering that these properties create class restrictions between domains and ranges. The complete list of concept properties, with their domains and ranges used in our ontology is shown as in Table 1.

Table 1. Domains and Ranges of Class Properties

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Domain</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>hasAtomic</td>
<td>Page</td>
<td>AtomicUnit</td>
</tr>
<tr>
<td>hasType (subproperty of hasAtomic)</td>
<td>Page</td>
<td>Type</td>
</tr>
<tr>
<td>hasChild</td>
<td>AtomicUnit</td>
<td>AtomicUnit</td>
</tr>
<tr>
<td>hasPage</td>
<td>AtomicUnit</td>
<td>Page</td>
</tr>
<tr>
<td>hasParent</td>
<td>AtomicUnit</td>
<td>AtomicUnit</td>
</tr>
<tr>
<td>hasPosition</td>
<td>AtomicUnit</td>
<td>Page</td>
</tr>
<tr>
<td>hasPreference</td>
<td>User</td>
<td>Preference</td>
</tr>
<tr>
<td>hasRating</td>
<td>AtomicUnit</td>
<td>Rating</td>
</tr>
<tr>
<td>hasRequest</td>
<td>User</td>
<td>Request</td>
</tr>
<tr>
<td>hasSession</td>
<td>User</td>
<td>Session</td>
</tr>
</tbody>
</table>

The overall relationship of classes through class properties that has been used in this ontology can be generalized as follows:

- Every atomic unit belongs to a certain page. This relationship is presented through hasAtomic property. This property gives the possibility of synthesizing a page “on-the-fly” based on atomic units.
- Each atomic unit can have a link from whom it can be referenced and to whom it can reference. This relationship is represented through hasParent and hasChild property.
- Atomic units can have a certain position within a page. The position would involve the right, left, center, header or footer region of the page depending on the adaptation scale. This relationships is described through hasPosition property.
- Atomic units can have importance within a page concerning a certain concept. This importance has been emphasized through hasRating property which relates AtomicUnit class to some Rating subclasses such as High, Medium or Low.
- Finally, three properties have been defined concerning the user domain that creates relationships such as:
  - User can have a request in a web page. The request can be a complete page or an atomic unit and furthermore these requests can be gathered into sessions. These relationships are described with hasRequest and hasSession property.
  - And finally users can have a browsing preference which are being described by hasPreference property.

3.3. Specifying Concept Instances

Instances represents individuals that belong to a class, these are like class members. We have defined seven instances for the user class with the purpose of assigning them different preferences to see the inference on the ontology.

4. Rules and Inference in the Ontology

We have mentioned above that class properties created relationships between concepts. This relationship can be represented by predicate logic[1]. If we would use the class properties to link domain and range then we can create several base predicates in the form:

\[
\text{property}(X_i,Y_j)
\]

where both \( X_i = \{x_1, x_2, ..., x_n\} \) and \( Y_j = \{y_1, y_2, ..., y_n\} \) represent a domain and range of a class. From here we can create the following base predicates for our ontology:

\[
\text{hasAtomic}(X_{Page}, Y_{AtomicUnit})
\]
\[
\text{hasChild}(X_{AtomicUnit}, Y_{AtomicUnit})
\]
\[
\text{hasParent}(X_{AtomicUnit}, Y_{AtomicUnit})
\]
\[
\text{hasPage}(X_{AtomicUnit}, Y_{Page})
\]
\[
\text{hasPosition}(X_{AtomicUnit}, Y_{Page})
\]
\[
\text{hasPreference}(X_{User}, Y_{Preference})
\]
\[
\text{hasRating}(X_{AtomicUnit}, Y_{Rating})
\]
\[
\text{hasRequest}(X_{User}, Y_{Request})
\]
\[
\text{hasSession}(X_{User}, Y_{Session})
\]

Having these base predicates, we can further infer by adding other rules. For example to express that an atomic unit can belong only to
one page, have only one rating (low, medium or high) and can have exactly one position within a page we can write as follows:

\[
\text{atomicUnit} \leftarrow \text{hasPage}(X_{\text{AtomicUnit}}, Y_{\text{Page}} = 1), \\
\text{hasPosition}(X_{\text{AtomicUnit}}, Y_{\text{Position}} = 1), \\
\text{hasRating}(X_{\text{AtomicUnit}}, Y_{\text{Rating}} = 1)
\]

Defining a page that might have some type of atomic units and some certain number of it, can be written as:

\[
\text{Page} \leftarrow \text{hasType}(X_{\text{Page}}, \exists Y_{\text{Type}}), ... \\
\text{hasAtomic}(X_{\text{Page}}, \max(Y_{\text{AtomicUnit}} = 20))
\]

We have specified the maximum number of atomic units per page to 20 which is more than enough.

Other rules are as follows:

\[
\text{Preference} \leftarrow \text{hasRating}(X_{\text{AtomicUnit}}, Y_{\text{Rating}} = 1)
\]

\[
\text{Rating} \leftarrow \text{hasRating}(X_{\text{AtomicUnit}}, Y_{\text{Rating}} = 1)
\]

\[
\text{User} \leftarrow \text{hasPreference}(X_{\text{User}}, \exists Y_{\text{Preference}}), ... \\
\text{hasSession}(X_{\text{User}}, \exists Y_{\text{Session}})
\]

\[
\text{Request} \leftarrow \text{hasRequest}(X_{\text{User}}, Y_{\text{AtomicUnit}} = 1) \lor ... \\
\text{hasRequest}(X_{\text{User}}, Y_{\text{Page}} = 1)
\]

### 4.1. Inference in the ontology

To test the inference in our ontology we have created new concept class called Visits with two subclasses named NoviceVisits and ExperiencedVisits respectively. The rules that we have used for these two classes are that novice visitors do not have a particular interest in the topic but they browse randomly and the number of requests is not very big. From average five requests[8] per session they visit maximum two or three. Because they are unexperienced visitors, atomic units should be putted in different positions in a page as stated in the Position subclass of AtomicUnit class, to address their needs. This is helpful in avoiding the “lost in hyperspace” effect.

The rules used for Novice Visits are:

\[
\text{Visits} \leftarrow \text{hasPreference}(X_{\text{User}}, Y_{\text{Preference}} = 3) \land \text{hasSession}(X_{\text{User}}, Y_{\text{Request}} = 3)
\]

\[
\text{Visits} \leftarrow \text{hasPosition}(X_{\text{AtomicUnit}}, \exists Y_{\text{Page}}) \\
\text{hasRating}(X_{\text{AtomicUnit}}, \forall Y_{\text{Rating}} = \text{HighRating})
\]

where:

\[
\exists Y_{\text{Page}} = \text{LeftPosition} \lor \text{Middle} \lor \text{Header} \lor \text{Footer}
\]

The experienced visits on the other hand tend to be more specific with well defined preferences that does not overpass two preferences per visit and the session tend are more long with an average of 5 requests per session [8]. The rules used for experienced visits are:

\[
\text{Visits} \leftarrow \text{hasPreference}(X_{\text{User}}, Y_{\text{Preference}} = 2) \land \text{hasSession}(X_{\text{User}}, Y_{\text{Request}} = 3) \land \max(X_{\text{User}}, Y_{\text{Request}} = 5)
\]

\[
\text{Visits} \leftarrow \text{hasPosition}(X_{\text{AtomicUnit}}, \exists Y_{\text{Page}}) \\
\text{hasRating}(X_{\text{AtomicUnit}}, \forall Y_{\text{Rating}} = \text{LowRating}, \text{MediumRating})
\]

where:

\[
\exists Y_{\text{Page}} = \text{RightPosition} \lor \text{Middle}
\]

The inference testing of these rules has been done by using Protégé’s Fact++[13] inference engine. The Fact++ uses the FaCT system (Fast Classification of Terminologies) which represents a Description Logic Classifier for terminology classification. In OWL is being used for concept classification.

From Fact++ reasoning two main classification of concepts occur in our ontology. The first classification is that NoviceVisits and ExperiencedVisits appear under AtomicUnits class. Which is logical because in several rules for the visits above we have emphasized these visits through position and ratings of atomic units within a page as well as users preferences. We can conclude that the reasoner creates two types of atomic units in a page for two types of users.

The second classification is the one that happens under User class. We can see that under this class we have ExperiencedVisits, NoviceVisits and Preference listed which we did not have in our original ontology. The reasoner inferred to this conclusion based on the rules that
we have given above concerning the request that had some atomic units which had some rating.

The inferred classification classes can be seen below in Fig. 4.

![Inferred model of the Ontology](image)

**Figure 4. The Inferred model of the Ontology**

5. Conclusion and Future Work

In this paper we have presented a complete knowledge representation for Adaptive Web-Based Systems using ontologies. The ontology used comply with the requirements for expressiveness, modularity and interoperability presented in Top-Down Five Layer framework. We have also created some rules and tested the ontology inference in an inference engine and we saw that findings comply and can be used as knowledge base for the second and third layer of our framework.

The future work would involve creating a data representation that will use the proposed ontology, finding the possibility of automated creation of content-concept linking, using adaptation algorithms and presenting a content in presentation layer. It is worth mentioning that the above tasks are not trivial and require the involvement of many disciplines like web mining, machine learning, semantic web and many other related fields.

References


