Enhanced Semantic Web Layered Architecture Model

Islam H. Harb, Salah Abdel-Mageid, Hassan Farahat, M. S. Farag
Al-Azhar Univ. Al-Azhar Univ. Al-Azhar Univ. Al-Azhar Univ.
Egypt. Egypt. Egypt.
harbhany@yahoo.com, salah.ramadan@yahoo.com, Farag.sayed@yahoo.com

Abstract: - This paper introduces the traditional web and its limitations, and how these limitations can be overcome by putting the lights on a new interesting approach for the web which is the Semantic Web. The most common technologies that can be used to construct such smart web are discussed briefly, and then its current layered architecture models as proposed by Tim Berners-Lee and others are evaluated to alleviate discrepancies and weak points. An enhanced architecture obeying layered architecture evaluation criteria and standard principles is proposed. This enhanced model is evaluated and contrasted against other models.

Key-Words: - Semantic web, ontology, XML, RDF, URI and Functional Layers.

1 Introduction

The Semantic Web may be considered as an evolution to this WWW which aims to make all the information and application data on the internet universally shared and machine processable in a very efficient way. It is an intelligent web which can understand the information semantics and services on the Internet by applying technologies and enabling inference rules to increase users’ satisfaction while searching the web content [17, 19].

Motivations for inventing the Semantic Web are the limitations of the traditional Web [1] where Semantic Web provides solutions for these drawbacks by offering more efficient technologies such as XML, RDF and Ontology.

The Semantic Web was introduced by Tim Berners-Lee who invented the traditional World Wide Web [2, 4]. Four versions of Semantic Web architecture were proposed. Such versions describe the languages needed for data interoperability between applications in the form of layering architecture where each layer represents a language that provides services to the upper layer. However, layers described in such versions suffer from several deficiencies such as poor abstraction and rarely functional descriptions. Gerber [12] avoided those deficiencies and designed a new architecture. Additional layer called “Rules” layer was added although its functionality is already embedded inside other layers. The authors in [18] explored many layers compared to the Gerber’s model. However, the overlapped functionalities among layers are more than overlapping in the Gerber’s model and this increases the difficulty of the system engineering description and the layers integration.

This paper introduces an Enhanced Semantic Web Layered Architecture Model. This enhanced model introduces a new architecture which follows the layered architecture as [12], [18]. This enhanced proposed architecture alleviates the previous work weak points. Some layers are merged, and a novel vision for the “Identity Verification” layer is suggested.

This paper is organized as it follows. Section 2 discusses the traditional Web limitations. The Semantic Web technologies are presented in Section 3. Section 4 evaluates the basic Semantic Web architecture. The enhanced proposed architecture model is introduced in Section 5. Finally, Section 6 concludes this work.

2 Traditional Web Limitations

The content of the World Wide Web may be classified into documents and data [1], where documents are everything readable by the human such as reports and mails while data may be processed by machines to make them readable and can be handled by the human.

The Hypertext Markup Language (HTML) is a language to specify the structure of documents for retrieval across the Internet using browser programs of the World Wide Web. It was designed to create web pages and to display the data on these web pages. It doesn't focus on what the data is or how the data can be stored and transported in an efficient way. The HTML is used in the traditional web as a language or tool that concentrates on the design of the web page and how this page looks like. It is one of the main technologies used in the traditional web.
It uses Meta data tags to describe information about data. HTML files have a basic structure that we have to work within.

```html
<html>
<head>
<title>MY HOME PAGE NAME</title>
</head>
<body>
THIS IS WHERE THE BODY OF THE WEB PAGE GOES (WHAT WE SEE IN NETSCAPE/INTERNET EXPLORER).
</body>
</html>
```

The text/bracket combinations are called tags. Note they come in pairs. There is always a beginning tag (`<title>`) and an end tag (`</title>`). The beginning tag signals the web explorer that a new tag/task is starting. The end tag tells the web explorer that the tag/task has ended.

It is a human readable but cannot be processed by the machine efficiently. It does not provide any semantics on the data or its nature, as it is just used for drawing and displaying this data on the web pages. It is very difficult to represent data that holds complex relationships between its elements in an efficient way to be understood by the machine.

### 3 Semantic Web Technologies

The Semantic Web takes the solution further. It involves publishing in languages specifically designed for data: Resource Description Framework (RDF), Web Ontology Language (OWL), and Extensible Markup Language (XML) [1]. In contrast with the simple HTML, the XML allows content creators to label information in a meaningful way [2] (for example `<Car><Model>1981</Model></Car>`) but the machine still knows nothing about what is meant by this structure. The RDF then comes to represent the relationships between the data items and give more meaning for the XML labels.

The OWL provides the semantics for the data syntax and solves some problems occurred such as “two databases may use different identifiers for the same concept” [2], then the machine needs OWL to discover the data which have the same meaning. These expressive technologies enable the data to be represented in a machine readable structure and hence enable the machine applying inference rules on the data to obtain meaningful results to improve the search process. In the next subsections we introduce these Semantic Web technologies.

#### 3.1. Uniform Resource Identifier (URI)

It is used to identify resources on the web, in which every resource in the World Wide Web should be uniquely identified so we give it a URI. Resources could be anything such as a book, document, or video. There are different forms of the URIs. The most familiar form is the URL (Uniform Resource Locator) which is typed in the Web browser to locate its corresponding resource, so it has two functionalities which are identifying and locating the resources. There are other forms that only identify the resources but can’t tell us their locations.

Because the Web is too large to be controlled by only one person or organization [9], so creation of URIs is decentralized and anyone could create URI for her resources. It is clear that a problem of identifying the same resource with more than one URI may exist, but it is the cost of having such flexible and simple technique to identify resources on the Web.

#### 3.2. Extensible Markup Language (XML)

XML is the most popular and simplest technology that is used to send documents on the Web and it is readable by both the human and the machine. This machine readability is one of the features that make XML very powerful technology. XML documents are consisted of elements where each element is enclosed between two tags, one is the opening tag and the other is the closing tag and between these two tags there may be other elements as children of this enclosing parent element (hierarchal structure).

Now we are going to give an example for the XML syntax:

```xml
<Sentence>
    <human href="http://www.myself.com/">I</human>
    want to be a <human>doctor</human>.
</Sentence>
```

A parent tag “Sentence” has enclosed two other elements which are “human ⇒ I “and “human ⇒ doctor” . They are connected by the phrase “want to be a”, while the element “I” has an attribute (href="http://www.myself.com/") giving more description about it. The Sentence is an element inside an XML document to indicate a normal sentence (I want to be a doctor) in a readable way by machine programs.

As the XML is very flexible, it enables anyone to create her own format and use her own words and vocabularies. It is then possible that two different persons use common words such as “doctor” but with two different meaning like saying doctor referring to professor in the university or referring to physician. So XML Namespace comes to solve this
problem where any item defined in a certain namespace should be unique within this namespace. So when an item is used as a tag name in an XML document, its namespace may be referred to prevent the confusion that could occur. Now we give the same example using the XML Namespace concept [9]:

```xml
<Sentence xmlns="http://example.org/xml/documents/"
xmlns:p="http://person.example.net/xmlns/">
    I want to be a <p:human>doctor</p:human>.
  </p:human></Sentence>
```

There is a default namespace for any tag mentioned without its namespace. In this example, “http://example.org/xml/documents/” is the default namespace for the tags not preceded by the “p:”.

### 3.3 Resource Description Framework (RDF)

Resource Description Framework (RDF) is a foundation of metadata processing. It provides interoperability between applications that exchange machine-understandable information on the Web [5, 7] and it defines the relationship between the resources on the web. There are different syntaxes that can represent the RDF; one of the most popular syntax is the XML where the RDF based on this syntax is called RDF/XML model. The RDF statement is written in a triple form consisting of three parts which are the subject, the predicate and the object, so it seems like it is a natural phrase but its parts are URIs as they are resources on the Web. Consider the following example [5]:

```
http://www.w3.org/Home/Lassila has creator Ora Lassila
```

Where the subject is “http://www.w3.org/Home/Lassila”, the predicate is “has creator” and the object is the literal “Ora Lassila”.

There are two XML syntaxes to represent the RDF data model which are serialization syntax and the abbreviated syntax. The above sentence can be represented in RDF/XML model using the serialization syntax as it follows.

```xml
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://description.org/schema/">
  <rdf:Description about="http://www.w3.org/Home/Lassila">
    <s:Creator>Ora Lassila</s:Creator>
  </rdf:Description>
</rdf:RDF>
```

RDF defines a simple, yet powerful model for describing resources. A syntax (which is XML) representing this model is required to store instances of this model into machine-readable files and to communicate these instances among applications. RDF imposes formal structure on XML to support the consistent representation of semantics [7].

### 3.2. Ontology

The term "ontology" can be defined as an explicit specification of conceptualization [6, 8]. The conceptualization means modeling certain domain and the Ontology is used to describe important concepts of this domain, so it is the specification of this conceptualization. Ontology is the stage where the vocabularies related to a specific domain should be defined. It provides the capability to make analysis on the relationships between the vocabularies to discover problems such as the existence of two vocabularies of the same meaning. In this stage the relationships between vocabularies of a specific domain are created in hierarchal form by using the inheritance and classes concepts. Languages such as OWL (web ontology language) which may be considered as a syntactic extension for RDF/RDFS are provided at this stage.

### 4 The Semantic Web Architecture

Tim Berners-Lee proposed four versions of Semantic Web architecture [15]. These versions are shown in Fig. (1) As V1 to V4. These versions describe the languages needed for data interoperability between applications in the form of layering architecture where each layer represents a language that provides services to the upper layer. The four versions (layered architectures) are composed mainly of seven functions. Each function is nearly represented by a layer. As we are talking about architecture concept then there are some discrepancies and irregularities in Tim Berners-Lee’s architectures considering the layered architecture evaluation criteria [11, 12]. The layered architecture evaluation criteria are clearly defined context, an appropriate level of abstraction, hiding of unnecessary implementation details, clearly defined functional layers, appropriate layering, and Modularity. These criteria are used as an evaluation for any layering architecture. Table 1 summarizes the conformance or non-conformance of the Semantic Web layered architectures to the criteria. It also describes a comparison between these models and our enhanced model that will be discussed in the next section.
4.1. Clearly Defined Context
The context of the Semantic Web layer architecture is the languages necessary for implementing the Semantic Web that provides a way for data interoperability between applications. These languages should be readable and can be processed by the machines. The current Semantic Web layered architecture conforms to this criterion.

4.2. Appropriate Level of Abstraction
The layered architecture should not disclose many details and information; it should be abstracted enough where only the functionality is represented without going deep into the details. The top three layers have enough level of abstraction and define functionality, but the rest of the layers specify unnecessary details about technologies rather than functionalities. The current Semantic Web layered architecture does not conform to this criterion.

4.3. Hiding of Implementation Details
Referring to Fig. 1, it is clear the first three bottom layers indicate how we can implement the functionality as they represent the technologies and implementation issues. The current Semantic Web layered architecture does not conform to this criterion.

4.4. Clearly Defined Functional Layers
The bottom layers such as URI, XML and RDF represent the technologies while they should depict the functionality, so they are not clearly defined functional layers. So the Semantic Web layered architecture does not conform to this criterion.

4.5. Appropriate Layering
This criterion is concerned of interlayer interfaces and dependencies. Architecture should be designed such that the services provided by each layer to its upper one are well known. The current Semantic Web layered architecture does not give us this information about what are the requirements of an upper layer with respect to its lower one, so it does not conform to this criterion.

4.6. Modularity
This criterion depends on the functionality defined by each layer in the architecture. There are layers in the Semantic Web architecture that do not represent functionality, so the Semantic Web layered architecture does not conform to this criterion.

5 Enhanced Semantic Web Layered Architecture Model (ESLAM)
ESLAM is an enhanced semantic web layered architecture taking Tim Berners-Lee’s architectures as a reference with the consideration of the layered architecture evaluation criteria in addition to the principles of “The Seven Layers of the OSI Architecture” [13]. Such principles are illustrated as it follows.

- Do not create so many layers as to make difficult the system engineering task describing and integrating these layers.
- Create separate layers to handle functions which are manifestly different in the process performed or the technology involved.
- Collect similar functions into the same layer.

Figure 1: The Four Versions of Semantic Web Architecture Proposed By Tim Berners-Lee

Figure 2: A. J. Gerber, A. Barnard and A. J. van der Merwe Architecture
Gerber [12] designed a new architecture considering the layered architecture evaluation criteria which is given in Fig. 2, but it still suffers from weak points. Our enhanced model alleviates the Gerber architecture weak points. To understand the first weak point in Gerber architecture, we have to know exactly the functionality of both the ontology and the logic layers. Ontology is used to describe the meaning and relationships of terms. This ontology description (in RDF, of course) helps machines to use terms more easily. It helps machines to decide how to convert between them and it can be used as rules to infer new data from data already known [9]. An example of this inference type using DAML+OIL (an ontology language) is as it follows. DAML construct that we run through is the daml:inverseOf property. Using this property, we can say that one property is the inverse of another [3].

```
:hasName daml:inverseOf :isNameOf .
:Microsoft_Manager :hasName "Bill Gates"
Implies:
"Bill Gates" :isNameOf :
```

Microsoft_Manager

Logic is the layer responsible for stating any logical rules and it allows the agent or the machine from making useful inference. This layer function may be based on the first order predicate logic [14]. So there is no need to have an additional layer called “Rules” layer because its functionality is already embedded inside both of the ontology and logic layer. If we refer to version one of Tim Berners-Lee’s architecture, we find that there is no layer that can be mapped to “Rules” layer and if we refer to version three, we find it at the same level of the OWL and of the same color which indicates that it is of the same functionality and can be supported by “ontology” layer.

From Fig. 3 [15], we can see that both “Logic Framework” and “Rules” are equivalent to the same functionality of providing logical rules based on the first order logic to help in the inference process, so this indicates that there is no need to make them two separate layers. Following the principles of “The Seven Layers of the OSI Architecture”, our proposed model collects the same functionality into the same layer. Consequently, in our model, the “Rules” layer will be embedded within the “Logic” layer only or within both the “Ontology” and the “Logic” layers. The second weak point is related to the vertical layers, where the “Identity Verification” supports only the first four layers (from the “Unique Identification Mechanism” to “Ontology”) but this is not enough. To understand the reason why the Identity Verification should be extended, we have to get familiar with the functionality of the “Trust” layer.

Trust is the layer that is responsible for evaluating the trustworthiness of the information. This evaluation is subjectively done by information consumers based on some trust mechanisms and policies. One of these trust mechanisms is called “Context-Based Trust Mechanisms” [16] that use meta-information about the circumstances in which information has been claimed, e.g. who said what, when and why. So we need the “Identity Verification” mechanism to be supported horizontally from the “Unique identification Mechanism” up through the “Proof” layer. As we know that this architecture is a layered architecture, the trustworthiness of each layer depends on that of all the lower layers.

There is a third weak point that the architecture has the same triangular form problem as in Tim Berners-Lee’s one, where there are no clear reasons explaining why the layers are of different sizes. It is not clear if it means that the narrow upper layer uses only part of the functionality of its lower wider layer or not. The description of our proposed model follows.

Identify Web Resources. To identify items on the Web, we may use a uniform system of identifiers where each identified item is considered a “resource”. It is considered as a unique way for identifying objects in the Semantic Web and between different layers and the associated languages in the Semantic Web architecture. The URI can be used as a technology to achieve such functionality.

Syntax Description Language. It is used in the semantic web as a representation for the data that are uniquely defined in the lower URI layer. This representation is readable and can be processed by
the machine in an efficient way. The XML can be used as the technology to achieve such functionality.

Figure 4: The Enhanced Semantic Web Layered Architecture

Meta-data Data Model. It imposes structure that provides for the unambiguous expression of semantics and, as such, it enables consistent encoding, exchange, and machine-processing of standardized metadata. It can be considered as the first layer that provides semantics for the data by defining specific formats, models and syntax. RDF and RDFS can be considered as the technologies to achieve such functionality.

Ontology. The main role of this layer is to provide the semantics of specific domain. It defines the vocabularies for specific domain and then describes the relationships of these vocabularies. Ontology is an explicit specification of conceptualization, means that ontology is a description of the concepts and its relationships. In order to understand in communication, ontology must be shared. OWL and DAML+OIL can be considered as the technologies to achieve such functionality.

Logic and Inference. It is not implemented yet. It is supposed to be the layer that will make inference operations on the available data to obtain other related and useful information for the user. There is no technology for this functionality yet but there is a language that is called Rules Interchange Format (RIF) that can be very close representative for such technology used to achieve this functionality.

Proof and Trust. Proof is used to prove the correctness of the information extracted and Trust is the layer that is responsible for trusting the semantic web information depending on the source of this information and the policies that allow or prevent trusting certain sites.

Figure 4 depicts ESLAM as our proposed Enhanced Semantic Web Layered Architecture. This architecture follows the layered architecture evaluation criteria and it also conforms to the principles of “The Seven Layers of the OSI Architecture” to alleviate the Gerber weak points. Our enhanced proposed architecture has no the “Rules” Layer and it has extended the “Identity Verification” vertical layer to support all the horizontal layer except for the Trust that uses this feature to evaluate the trustworthiness of the information before passing it to the application layer. The trust layer has the responsibility of making sure that the data is trusted by checking the digital signature to see who prepares the data, when and why. All the lower layers should apply digital signature. This makes the Identity verification vertical layer extending from the lowest layer up to the proof layer. The Identity verification vertical layer may not extend to the trust layer because the trust layer does not apply digital signature on its role as there is no upper layer to verify that digital signature. The Trust layer then verifies all the digital signatures that are added by the lower layers but never puts its own digital signature.

<table>
<thead>
<tr>
<th>Layered architecture evaluation criteria</th>
<th>Tim’s models Conformance</th>
<th>Enhanced model conformance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clearly defined context</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Appropriate level of abstraction</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Hiding of implementation details</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Clearly defined functional layers</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Appropriate layering</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Modularity</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

6 Conclusion

The limitations of the conventional Web and how the Semantic Web overcomes those limitations have been discussed. Then we discussed the common technologies used to construct the Semantic Web and we presented the four architectures proposed by Tim Berners-Lee. We evaluated these architectures according to layered architecture evaluation criteria. These four architecture versions as a reference and the principles of a layering architecture criteria. These four architecture versions as a reference and the principles of a layering architecture criteria. These four architecture versions as a reference and the principles of a layering architecture criteria. These four architecture versions as a reference and the principles of a layering architecture criteria.
References: