

TOWARDS A SEMANTIC WEB LAYERED ARCHITECTURE

A. J. Gerber
Meraka Institute
CSIR
Pretoria, South Africa
agerber@csir.co.za

A. Barnard
School of Computing
UNISA
Pretoria, South Africa
barnaa@unisa.ac.za

A. J. van der Merwe
School of Computing
UNISA
Pretoria, South Africa
vdmeraj@unisa.ac.za

Abstract

Tim Berners-Lee proposed a layered architecture for the languages of the Semantic Web in 2000 and suggested an adapted architecture in 2003. We evaluated the architecture according to a list of layered architecture characteristics and in this paper, based on aforementioned evaluation, we propose an adapted layered architecture for the languages of the Semantic Web based on fundamental aspects for layered architectures in information systems.

Keywords: Software architecture, Semantic Web, Web design, software evaluation

1 Introduction

The Internet revolutionised the way by which our society disseminates and uses information. The popularity of the Internet resulted in an exponential growth regarding the amount of information available on the Internet and the associated Web [33, 41]. A consequence of this popularity is an overload of information available on almost any topic. This overload creates several new problems for users. For example, it is increasingly difficult to find applicable information or to verify the origin of information. In addition it is difficult to verify that the information, or even the source of the information, is trustworthy [22, 35, 41].

The Internet was originally designed as an information space usable for both *humans* and *machines* [7]. In contrast, the current Web (as an Internet application) is a publishing mechanism for information targeted mainly at *human* users. It is thus possible to speculate that this results in the problems pertaining to the expedient access of appropriate and trustworthy information as mentioned. The current human users of the Web often do not have the means to solve these problems.

In 2001 Berners-Lee, Hendler and Lassila [11] presented a vision of a Web that is an information space usable by, in particular *machines*, the so called *Semantic Web*. Instead of attempting to process and manipulate Web information, a user would have a personal *agent* on his/her computer that would solve problems related to information overload, acquisition and discrepancy resolution [19]. The agent executes the first level of information management and the user only accesses or manipulates results. The proposed Semantic Web is at present

an international research effort with the goal to make Web content available for intelligent knowledge processing [14, 15, 18, 20, 21, 26, 34, 36, 39, 38, 44, 47]. The concept of the Semantic Web has captured the imagination of Web users, as well as the interest of academia and industry even though several technological issues still have to be resolved.

As part of the development of the Semantic Web vision, Tim Berners-Lee proposed a language tower or *layered architecture* for the Semantic Web in 2000 [8]. He proposed an adapted subsequent architecture in 2003 [9]. These architectures are depicted in Figures 1 and 2 respectively with added 'Layer' captions for reference purposes.

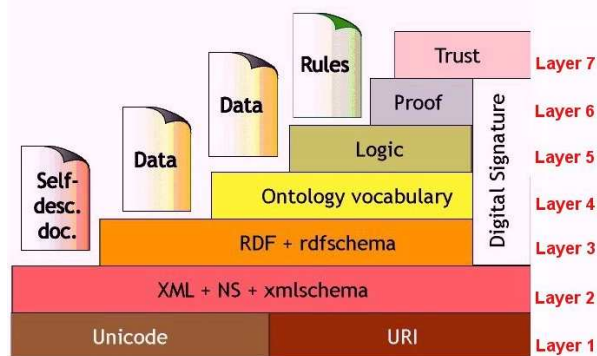


Figure 1. Berners-Lee's Original Semantic Web Architecture (reproduced from [8])

Several Semantic Web authors refer to and adopt one or both of these architectures in their research [1, 22, 23, 28, 32, 38, 40, 41, 46]. However, a concern about these architectures is that there is no concise description of the intended meaning. Furthermore, according to the characteristics of layered architectures, there seems to be inconsistencies on different levels. For example, as both functionality and technology are portrayed by the architectural layers, it is not clear what is depicted by Figures 1 and 2. In particular, certain layers specify a functionality required at that layer in the stack (such as *Ontology*), whilst other layers depict a technology (such as *XML*) without an explanation of the functionality that the specific technology embodies. It is also not clear what is intended by the stacking of the

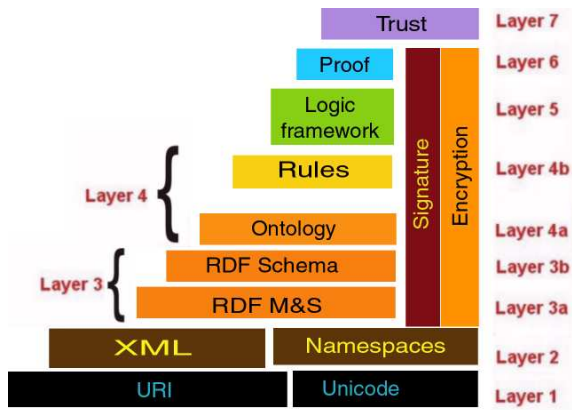


Figure 2. Berners-Lee's Adapted Semantic Web Architecture (reproduced from [9])

layers or the triangular structure. In addition, there appears to be more than one technology on certain layers (Layer 1 in Figure 1 as well as Layers 1 and 2 in Figure 2) and both architectures depict vertical layers (such as *Digital Signatures* in Figure 1 and *Signature* and *Encryption* in Figure 2). The intention behind this deviation from the structure is not reflected in literature.

It is not unreasonable to speculate that the architecture of any information system is the central aspect to consider during the design and implementation thereof. It is therefore plausible to state that the proposed architecture of the Semantic Web is crucial to its eventual realisation and that it is consequently necessary to attach undisputable meaning to the specification of the architecture for the languages of the Semantic Web.

A layered architecture for the Semantic Web that adheres to the fundamental aspects of layered architectures will assist in the development of Semantic Web specifications and applications. In addition, several of the current research and implementation issues associated with the implementation of the Semantic Web could potentially be resolved.

The purpose of this paper is to investigate the layered architecture of the Semantic Web at present, to comment on the layered structure thereof and to suggest adaptations that would reflect principles of layered architectures. We first discuss the concept *architecture* and specifically what a *layered architecture* represents within the information systems domain (sections 2.1 and 2.2). This is followed by a list of evaluation criteria for layered architectures in section 2.3. We apply the measurement criteria to the currently proposed Semantic Web architecture and report our findings in section 3. As a result of this evaluation, we suggest an adapted layered architecture in section 4. We reflect on some of the implications of the adapted architecture in section 5 and conclude in section 6.

2 Design and Evaluation Criteria for Layered Architectures

The term *layered architecture* is often used in the vernacular of IT researchers and system architects in the information systems application domain without a concise definition of its meaning [11, 27, 31]. In literature, a layered architecture is described only as an *architectural pattern* or *architectural style* [6, 43].

In this section we give a brief description of the concepts *software architecture* and *layered architecture*. In addition we provide a summary of a list of criteria for layered architectures identified previously [25] to which layered architectures should adhere.

2.1 Software Architecture

In the design and implementation of user requirements, the architecture of a software system is an indispensable mechanism [6]. The term *architecture* seems to defy the creation of a common, agreed upon definition within the information systems application domain. The abstraction of software system functionality into modules, together with its interfaces is in essence *the determination of the architecture of the system*. There is consensus among researchers and system architects that the determination of the architecture of a system is crucial to the successful understanding and development thereof, especially when the system envisaged is intricate and multifaceted. This is applicable to the description of the Semantic Web.

The concept *architecture* is commonplace in discussions of software engineering topics, but agreed-upon design and evaluation criteria are lacking in literature. Such criteria are on the one hand valuable for the determination of system architectures during the design phase, and on the other hand they provide a valuable tool for the evaluation of already existing architectures [25].

From definitions in literature [6, 24, 30, 43], we maintain that an architecture is a description of the components that encompass a system. The description of the components should include its organisation or structure, its defining features or properties, as well as its relationships together with available interfaces that allow interaction with it. In addition to this definition we identify the following important aspects to consider with regard to software architectures in general:

- An architecture is usually defined within a certain *context*. This context defines the view of appraisal of the system. In addition, this context determines the components necessary to implement the system, the properties of these components as well as the relationships between components and external entities. (Context relates to the notion of multiple architectures or structures defined by views as introduced by [6], [24] and [30].)
- An architecture is a *model* of the system in the given context, where a model is an abstraction of a real-

world representation [2]. It is important to realize that a model provides a means to view *only* the significant aspects of the entire system at a certain level of abstraction.

- The progression of the design of architectural models caused some architectural recurrences to evolve. These are described as *architectural patterns* [6] or *architectural styles* [43]. Examples of the best known architectural patterns include, but are not limited to, the client/server architectural pattern [13, 30], Peer-to-Peer architectural pattern [13, 30], three-tier architectural pattern [13, 24, 30], and the *layered architecture* or *layers* pattern [30, 43].

From the above it suffices to note that *layered architecture* is regarded as an architectural pattern or style that organises architectural components into layers. It can thus be regarded as an instance of an architecture and has to conform to the definitions of an architecture. In addition, layered architectures as a pattern provide best-practice solutions to recurring problems.

2.2 Layered Architecture

As discussed, the term *layered architecture* is also often used in the vernacular of IT researchers and system architects in the information systems application domain without a concise definition of its meaning [4, 5, 11, 25, 27, 31].

One of the best examples of layered architectures is the definition of network protocols found in the ISO/OSI model. This model defines all the methods and protocols required to connect computers by means of a network [54]. It separates the methods and protocols needed for network connectivity into seven different layers and each higher layer relies on services provided by a lower-level layer [13, 27, 37, 42].

Several other examples of layered architecture usage exist. The ISO/OSI network model comprises a *formal specification*. In contrast, the meaning of layered architectures in most other cases is implied. For example, Simpson uses a layered architecture as a top-level organisation to describe different viewpoints of real-time distributed computing systems [45], and Cruz and Xiao [17] use a layered approach to assist with data interoperability on the Semantic Web.

Arguably, layering is a common best-practice pattern used by software architects to break apart complex systems. In a layered architecture, the principal elements or components are arranged in the form of a layered cake where each layer rests on a lower layer. Generally, a layer represents a grouping of elements that provides related services. A higher layer might either use services only defined by the immediate lower layer (closed architecture) or services by all the lower layers (open architecture). However, the lower layers are unaware of higher layers [3, 13, 24, 37], and are not allowed to access functionality provided by upper layers. This implies a strict ordering of access to the functionality provided by components in a layered architecture in

one direction only [4].

Since a layered architecture is an instance of an architecture it is necessary to map the elements of a layered architecture to concepts discussed in the given architecture definition. The *layers* of a layered architecture map to components, or a grouping of components in an architecture (refer to section 2.1). In a layered architecture, the relationships and organisation of the architectural components are determined by the stacking and sequencing of layers, and are therefore depicted by the diagrammatical organisation of the layers from bottom to top.

2.3 Evaluation Criteria for Layered Architectures

The evaluation list for layered architectures comprises of the following six evaluation criteria:

- clearly defined context,
- an appropriate level of abstraction,
- hiding of unnecessary implementation details,
- clearly defined functional layers,
- appropriate layering, and
- modularity.

A more detailed list with a description and possible questions for evaluation is provided in an addendum in Table 2 at the end of this paper. The interested reader may also consult Gerber, Barnard and Van der Merwe [25].

3 The Evaluation of the Semantic Web Layered Architecture

The criteria for layered architectures are useful for the design of new architectures or the evaluation of existing ones. In Table 1 we summarise the adherence of the Semantic Web language architectures of Figures 1 and 2 using the criteria list from section 2.3.

Using the proposed criteria for the evaluation of layered architectures, we argue that the proposed Semantic Web layered architecture does not comply with the majority of the criteria. The layered architecture is specified within the context of the languages of the Semantic Web, but the architecture fails to conform to criteria with regard to abstraction, functionality description and interface specifications. The remainder of this section expands on the reasons for conformance or non-conformance of the Semantic Web layered architectures to the criteria.

3.1 Clearly Defined Context

The context is stated to be the languages required to implement the Semantic Web, or in other words, the languages required to describe the machine-processable information containing the meta-data descriptions of Web resources. The current Semantic Web layered architecture *conforms* to this criterion.

Table 1. Evaluation of the Semantic Web layered architecture

Criteria	Conforms	Does not conform
Clearly defined context	X	
Appropriate level of abstraction		X
Hiding of implementation details		X
Clearly defined functional layers		X
Appropriate layering (including well defined interfaces and dependencies)		X
Modularity		X

3.2 Appropriate Level of Abstraction

It is possible to argue that the whole Semantic Web language architecture is visible. However, it is appropriate and commendable to remove information from the model. The top three layers define functionality, but the rest of the layers specify *existing technologies* rather than functionalities. It is not clear what the function and interface to *Digital Signatures* as a vertical layer are, neither why *Unicode* and *URI* appear as two sections of the bottom layer. The current Semantic Web layered architecture *does not conform* to this criterion.

3.3 Hiding of Implementation Details

Referring to Figures 1 and 2, it is evident that the bottom three layers as well as *Digital Signatures* depict technologies or *implementation specifications* rather than the necessary functionalities required to implement the language architecture. In other words, these layers depict implementation details rather than functionality, and the current Semantic Web layered architecture therefore *does not conform* to this criterion.

3.4 Clearly Defined Functional Layers

The top three layers define functionality. It is not clear whether these are in relation to *languages* as the architecture context specifies, or applied functionality required for the implementation of the Semantic Web. The bottom layers specify existing technologies (such as *XML* and *RDF*) rather than the functions embodied by these layers. The function of *Digital Signatures*, *Unicode* and *URI* is also not clear from their positioning on the architecture. Hence, the Semantic Web layered architecture *does not conform* to this criterion.

3.5 Appropriate Layering

This criterion includes the specification of well defined interfaces and dependencies. The layering in the present Semantic Web architecture is not apparent. In particular, it is not clearly specified what the requirements of upper layers with regard to their lower layers are, nor is it possible to establish whether this is an open or closed architecture. The Semantic Web layered architecture *does not conform* to this criterion.

3.6 Modularity

We could argue that the conformance of the Semantic Web layered architecture with this criterion is *undefined* since the functionality of the layers has not been determined. Alternatively, if we need to indicate conformance or non-conformance, we could state that the Semantic Web layered architecture *does not conform* to this criterion.

4 Towards a Layered Architecture for the Semantic Web

We use the criteria discussed in section 2.3 as well as the reasons for the non-adherence presented in section 3, to propose adaptations to the Semantic Web architecture of Berners-Lee (Figures 1 and 2). The adapted architecture is presented in Figure 3. This model varies from the previously suggested architectures mainly because it abstracts and depicts related functionalities rather than the W3C technologies used to instantiate the functionalities.

In this section we discuss the adaptations according to the clarification of the architecture context, the abstraction of functionalities, the introduction of a reasoning layer and the development of a security stack.

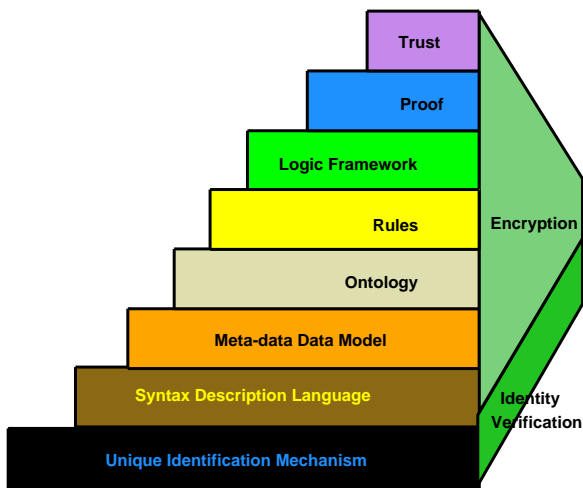


Figure 3. The adapted Semantic Web layered architecture comprises two orthogonal architecture stacks, the *language* stack and the *security* stack.

4.1 Clarification of the Architecture Context.

The architecture of a system defines the main components and their relationships. In order to describe a system with regards to its components and the relationships between the components, it is necessary to state the context of the architecture clearly. The context of the Semantic Web layered architecture is the *language functionalities* required to implement the Semantic Web. The emphasis should be on the *language functionality* in favour of the *language technologies* that could be used to implement such a functionality. In the architecture proposed in Figure 3, all the layers in both the *language* and *security* stacks represent functionalities.

4.2 Abstraction of Functionalities

All the layers in the architecture should portray the *functionalities* necessary to implement the language stack of the Semantic Web. At present, only the top three layers in both Figures 1 and 2 define functionality, the remainder of the layers specify *existing technologies* rather than functionalities. Abstracting the functionalities from the technologies, we propose (as a first iteration) that:

- *Unicode* and *URI* be replaced with *Unique Identification Mechanism*. *Unicode* aims to identify all the characters uniquely in all the written languages of the world [16] and *URI* (Uniform Resource Identifier) endeavours to identify any object uniquely with a character string [10]. Together, the functionality of these technologies could be described as the provision of a *unique identification mechanism* within the language stack for the Semantic Web.

- *XML*, *XML Schema* and *Namespaces* are depicted as *Syntax Description Language*. *XML* (Extensible Markup Language) provides a W3C standard for the exchange of data over various networks, especially the Web or WWW [53]. An XML schema is an XML document that defines the content and structure of one or more derived XML documents [49, 50, 51]. XML namespaces provides a simple method for qualifying element and attribute names used in XML documents by associating them with namespaces identified by URIs [12]. These technologies provide a syntax language for, among others, the envisioned Semantic Web.
- *RDF (M&S)* and *RDF Schema* are replaced by the *Meta-data Data Model* function. *RDF* (Resource Descriptive Framework) is a W3C Recommendation designed to standardise the definition and usage of meta-data with a simple data model. *RDF Schema* provides a predefined, basic type system for *RDF* models. *RDF Schema* extends *RDF* by giving an externally specified semantics to specific resources [48, 52]. This layer therefore provides a mechanism to model the meta-data required to implement the Semantic Web.

If we implement these changes, the criterion stating the *appropriate level of abstraction* will be adhered to because each layer depicts only the functionality required for that layer within the context of the languages for the Semantic Web. As a result, the system under review (the languages for the Semantic Web) can be viewed as a whole and only the aspects that are relevant for a specific layer are visible.

In a layered architecture, upper layers access the functionality of lower layers through interfaces. In order to specify these interfaces, the functionality and components within a layer have to be defined unambiguously. Once the functionality is specified, the definition of these interfaces will be a pre-requisite for the implementation of any Semantic Web language architecture. The specification of *clearly defined interfaces* is therefore one of the criteria that needs to be addressed in future research.

4.3 Development of a Security Stack

Security aspects do not play a *functional* role within, specifically, the *language* stack of our proposed Semantic Web architecture (Figure 3). We propose the development of a separate, parallel architecture to implement the security functionality required for the Semantic Web. However, this parallel security architecture needs to interface with the language architecture. In our proposed architecture we depict the security layered architecture as adjacent to the language architecture.

Figure 3 depicts an orthogonal, multi-dimensional architecture with different *views* (as introduced by [6], [24] and [30]). In addition, the structure of the layered architecture implies that the upper layers use only part of the func-

tionality defined by the interfaces of the immediate lower layer.

In order to be consistent with the definition of functionality components, we propose that *Digital Signatures* be replaced with *Identity Verification Mechanism*.

The original layered architecture of Berners-Lee depicts *Digital Signatures* and *Encryption* as vertical layers, and it might be argued that these *vertical* layers support the notion of an alternative view other than that of the language layers. Even though these technologies are not part of the *language* stack, they are indispensable for the eventual realisation of the Semantic Web. Refinement of the security architecture for the Semantic Web remains a topic for future research.

5 Implications of the Adapted Layered Architecture

From Figure 3 the reader may conclude that the proposed architecture with its adaptations depicts a simplification of the original architecture. The *abstraction* of the functionality of the original architectures does entail a simplification of the model. However, this abstraction enables the use of diverse and non-related technologies to implement the same functionality. In other words, different technologies could be used to instantiate the functionality of a specific layer in the proposed Semantic Web layered architecture.

The two-tower architecture of Horrocks, Parsia, Patel-Schneider and Hendler [29] (Figure 4) supports this notion. Horrocks et al. [29] present two possible instantiations of the Semantic Web layered architecture. We relate these instantiations to our adapted Semantic Web architecture in Figures 5 and 6.

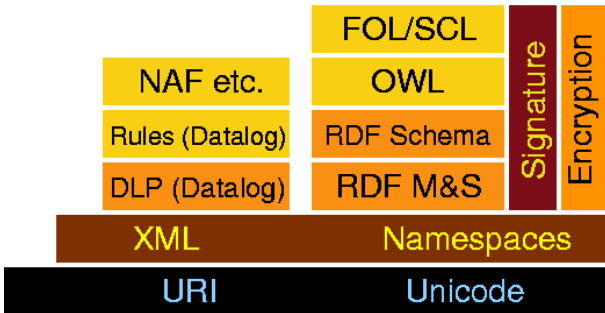


Figure 4. The two-tower architecture (reproduced from Horrocks et al. [29])

We note that in *tower 1* of Figure 4, Horrocks et al. omit an explicit data representation layer. DLP (Description Logic Programs) extends Datalog to include knowledge representation and Datalog was developed as a rule and query language for deductive databases and syntactically Datalog is a subset of Prolog, hence it does not constitute a data representation layer.

In tower 1 of Figure 4 the Ontology functionality is implemented by DLP. The Rules functionality is implemented with Datalog Rules and NAF (Negation As Failure). We depict this relationship in Figure 5.

In contrast to tower 1, tower 2 of Figure 4 implements the Meta-data Data Model functionality by means of RDF. The Ontology layer functionality is implemented using RDF Schema and OWL. We contend that the FOL/SCL layer provides for Rules and to a limited extent, Logic Framework functionality. We depict this instantiation in Figure 6.

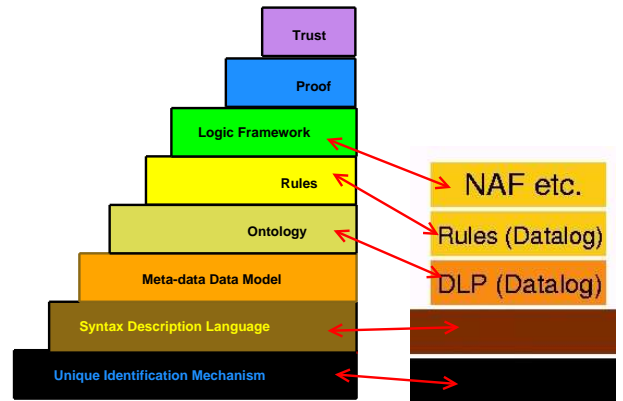


Figure 5. The two-tower architecture: tower 1 as instantiation of the SW architecture

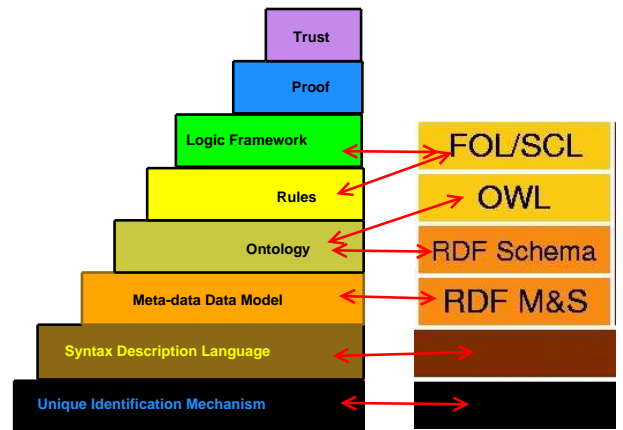


Figure 6. The two-tower architecture: tower 2 as instantiation of the SW architecture

We maintain that the current Semantic Web layered architecture of Berners-Lee [8, 9] was developed to depict technologies that were adopted as technology standards or recommendations by the W3C. This correlates with the mandate of the W3C that includes the development of *specifications* for the Web. However, we are of the opinion that the definition of a general, accepted layered architecture

with functional components might assist the W3C to develop diverse technology specifications for the implementation of the relevant functionalities. We furthermore maintain that this approach would *include* rather than *exclude* technologies that would have as benefit a more rapid adoption of the Semantic Web. The definition of W3C specifications needs an appropriate level of abstraction to be able to penetrate sufficiently into the implementation domain, and this is provided by our adapted layered architecture.

6 Conclusion

In this paper we propose a first iteration adaptation of the Semantic Web architecture adhering to the fundamental aspects of layered architectures within the information system domain. Using a previously compiled criteria list, we evaluate the present Semantic Web layered architecture of Berners-Lee. As a result, we propose an initial adaptation of this architecture.

We are of the opinion that a generalised layered architecture for the Semantic Web with well-defined functionalities will assist in the acceptance of diverse technologies for the implementation of the required functionalities. This will aid the eventual realisation of the Semantic Web. We acknowledge that all issues with regards to the layering of the Semantic Web languages have not been resolved by this approach. However, we are of the opinion that this is an important step towards realising the notion of the Semantic Web.

7 Addendum

Table 2 (reproduced from [25]) categorises the main characteristics of a layered architecture into criteria and presents a short description of each. In addition, questions that could be asked for evaluation purposes are included.

References

- [1] ANTONIOU G. AND VON HARMELEN F. *A Semantic Web Primer*. The MIT Press **2004**.
- [2] AVISON D. AND FITZGERALD G. *Information Systems Development: Methodologies, Techniques and Tools*. 3rd edition. McGraw-Hill **2003**.
- [3] BACHMAN C. **Personal Chronicle: Creating Better Information Systems, with Some Guiding Principles**. *IEEE Transactions on Knowledge and Data Engineering* **1989**: pp. 17–32.
- [4] BACHMANN F., BASS L., CARRIERE J., CLEMENTS P., GARLAN D., IVERS J., NORD R. AND LITTLE R. **Software Architecture Documentation in Practice: Documenting Architectural Layers**. *Technical Report CMU/SEI-2000-SR-004*, Carnegie Mellon Software Engineering Institute **2005**.
- [5] BALAKRISHNAN H., LAKSHMINARAYANAN K., RATNASAMY S., SHENKER S., STOICA I. AND WALFISH M. **A layered naming architecture for the internet**. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM Press, New York, NJ, USA **2004**. ISBN 1-58113-862-8, pp. 343–352.
- [6] BASS L., CLEMENTS P. AND KAZMAN R. *Software Architecture in Practice*. Addison Wesley Professional **2003**. ISBN 0-321-15495-9. URL: <http://www.aw-bc.com/catalog/academic/product/0,4096,0321154959,00.html>.
- [7] BERNERS-LEE T. **The Semantic Web Road Map**. W3C Website **1998**. URL: <http://www.w3.org/DesignIssues/Semantic.html>.
- [8] BERNERS-LEE T. **Semantic Web - XML2000**. W3C Website **2000**. URL: <http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html>.
- [9] BERNERS-LEE T. **Standards, Semantics and Survival**. *SIIA Upgrade* **2003**: pp. 6–10.
- [10] BERNERS-LEE T., FIELDING R., IRVINE U.C. AND MASINTER L. **IETF RFC3986 - Uniform Resource Identifiers (URI): Generic Syntax**. Web Document **2005**. URL: <http://www.rfc-archive.org/getrfc.php?rfc=3986>.
- [11] BERNERS-LEE T., HENDLER J. AND LASILA O. **The Semantic Web**. *The Scientific American*, vol. 5(1) **2001**. URL: <http://www.scientificamerican.com/>

Table 2. Design and Evaluation Criteria for Layered Architectures

Criteria	Description	Possible questions for evaluation
Clearly defined context	The context used to analyse the system determines its important aspects, assisting in the identification of the main components required to realise the system, its properties, its organisation, as well as the relationships between components.	Is it possible to identify the context from the description of the architecture?
Appropriate level of abstraction	The architecture model should be at a sufficiently high level of abstraction so that the system or subsystem under review can be viewed as a whole. Only the aspects of the system that are relevant at a certain level of abstraction should be visible at that level.	Can the system within the context be viewed as a whole? Are there any components/ properties/ relationships in the architecture model that could be removed without losing important information at this level of abstraction?
Hiding of implementation details	This criterion supports the above criterion regarding the level of abstraction. Implementation details should be hidden in an architectural model.	Are any implementation details visible in the description of the components/ properties/ relationships/ structures of the architecture?
Clearly defined functional layers	This criterion relates to the determination of the architectural components and their grouping into the appropriate layers.	Does the layer description specify a <i>function</i> of the layer within the system? Is the layer's function clear from its description and position in the architecture? Could the layer be removed without compromising the integrity of the system?
Appropriate layering (including well defined interfaces and dependencies)	Appropriate layering includes well defined interfaces and dependencies. This criterion relates to the organisation of the layers. The layers must clearly build on one another and its relationships and dependencies should be distinguishable, where any layer only accesses layers below it. This criterion also includes the specification of dependencies or access rules between layers, which is used to determine whether the architecture is open or closed.	Do the layers clearly build on one another? Does a specific layer only require functionality defined by lower layers and not those of upper layers? Is it possible to determine whether the layered architecture is open or closed?
Modularity	Components and hence layers should be modular. It should be possible to change the implementation of a layer as long as interfaces and functionality remains the same.	Is it possible to replace the implementation of a layer with another implementation of the same functionality and interfaces without compromising the integrity of the layered architecture?

2001/0501issue/0501berbers-lee.html.

- [12] BRAY T., HOLLANDER D. AND LAYMAN A. **Namespaces in XML Recommendation - REC-xml-names-19990114**. W3C Website 1999. URL: <http://www.w3.org/TR/REC-xml-names/>.
- [13] BRUEGGE B. AND DUTOIT A.H. *Object-oriented Software Engineering using UML, Patterns, and Java*. 2nd edition. Prentice-Hall 2004.
- [14] BUTLER M.H. **Barriers to real world adoption of semantic web technologies**. *HP Labs Technical Report*, vol. HPL-2002-333 2002. URL: <http://www.hpl.hp.com>.
- [15] CHERRY S.M. **Weaving A Web of Ideas**. *IEEE Spectrum, Special RD Report* 2002.
- [16] CONSORTIUM U. **The Unicode Consortium**. Unicode Website 2004. URL: <http://www.unicode.org/>.
- [17] CRUZ I.F. AND XIAO H. **Using a Layered Approach for Interoperability on the Semantic Web**. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering (WISE03)* 2003.
- [18] DE BRUIJN J., LARA R., POLLERES A. AND FENSEL D. **OWL DL vs. OWL flight: conceptual modeling and reasoning for the semantic Web**. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*. ACM Press, New York, NY, USA 2005. ISBN 1-59593-046-9, pp. 623–632. doi:<http://doi.acm.org/10.1145/1060745.1060836>.
- [19] DECKER S., MELNIK S., VAN HARMELEN F., FENSEL D., KLEIN M., BROEKSTRA J., ERDMANN M. AND HORROCKS I. **The Semantic Web: The Roles of XML and RDF**. *IEEE Internet Computing*, vol. 4 2000: pp. 63–74. URL: <http://csdl.computer.org/comp/mags/ic/2000/05/w5063abs.htm>.
- [20] DORI D. **ViSWebthe Visual Semantic Web: unifying human and machine knowledge representations with Object-Process Methodology**. *The VLDB Journal*, vol. 13(2) 2004: pp. 120–147. ISSN 1066-8888. doi:<http://dx.doi.org/10.1007/s00778-004-0120-x>.
- [21] EUZENAT J. AND NAPOLI A. **The Semantic Web: Year One**. *IEEE Intelligent Systems*, vol. 1094-7167/03 2003.
- [22] FENSEL D. **Language Standardization for the Semantic Web: The Long Way from OIL to OWL**. In *Distributed Communities on the Web: 4th International Workshop, DCW 2002, Sydney, Australia*, vol. 2468 / 2002 2002, pp. 215–227.
- [23] FENSEL D. **Ontology-Based Knowledge Management**. *IEEE Computer*, vol. 35 no 11 2002: pp. 56–59. URL: <http://csdl.computer.org/dl/mags/co/2002/11/ry056.htm>.
- [24] FOWLER M. *Patterns of Enterprise Application Architecture*. Addison-Wesley 2003.
- [25] GERBER A., BARNARD A. AND VAN DER MERWE A. **Design and Evaluation Criteria for Layered Architectures**. In *MSVVEIS Workshop hosted at the 8th International Conference on Enterprise Information Systems, Paphos, Cyprus* 2006, pp. 163–172.
- [26] GRAU B.C. **A Possible Simplification of the Semantic Web Architecture**. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*. ACM Press, New York, NJ, USA 2004. ISBN 1-58113-844-X, pp. 704–713. doi: <http://doi.acm.org/10.1145/988672.988769>.
- [27] HALLBERG B. *Networking: A Beginner's Guide*. 2nd edition. Osborne / McGraw-Hill 2001.
- [28] HENDLER J. **Agents and the Semantic Web**. *IEEE Intelligent Systems*, vol. 16 2001: pp. 30–37.
- [29] HORROCKS I., PARSIA B., PATEL-SCHNEIDER P. AND HENDLER J. **Semantic Web Architecture: Stack or Two Towers?** *Lecture Notes in Computer Science: Principles and Practice of Semantic Web Reasoning: Third International Workshop*, vol. 3703 / 2005 2005. ISSN 0302-9743. doi: 10.1007/11552222.
- [30] JACOBSON I., BOOCH G. AND RUMBAUGH J. *The Unified Software Development Process*. Addison-Wesley 1999.
- [31] JECKLE M. AND WILDE E. **Identical Principles, Higher Layers: Modeling Web Services as Protocol Stack**. In *XML Europe 2004, Amsterdam* 2004. URL: <http://dret.net/netdret/docs/wilde-xml europe2004.pdf>.
- [32] KIFER M., BRUIJN J., BOLEY H. AND FENSEL D. **A Realistic Architecture for the Semantic Web**. *Lecture Notes in Computer Science*, vol. 3791 2005: pp. 17 – 29.
- [33] LYMAN P. AND VARIAN H.R. **How Much Information, 2003**. How Much Information Website 2003. URL: <http://www.sims.berkeley.edu/how-much-info-2003>.

- [34] MCGUINNESS D.L. **Question Answering on the Semantic Web.** *IEEE Intelligent Systems*, vol. Vol. 19, No. 1 **2004**: pp. 82–85. URL: <http://csdl.computer.org/dl/mags/ex/2004/01/x1082.htm>.
- [35] MCGUINNESS D.L., FIKES R., HENDLER J. AND STEIN L.A. **DAML+OIL: An Ontology Language for the Semantic Web.** *IEEE Intelligent Systems* **2002**: pp. 72–80. URL: <http://csdl.computer.org/dl/mags/ex/2002/05/x5072.htm>.
- [36] MILLER E. **The W3Cs Semantic Web Activity: An Update.** *IEEE Intelligent Systems*, vol. 1094-7167/04 **2004**: p. 95.
- [37] NUTT G.J. *Centralized and Distributed Operating Systems.* Prentice-Hall International Editions **1992**.
- [38] OBERLE D., STAAB S., STUDER R. AND VOLZ R. **Supporting application development in the semantic web.** *ACM Trans. Inter. Tech.*, vol. 5(2) **2005**: pp. 328–358. ISSN 1533-5399. doi: <http://doi.acm.org/10.1145/1064340.1064342>.
- [39] PALMER S.B. **The Semantic Web: An Introduction.** W3C Website **2001**. URL: <http://infomesh.net/2001/swintro/>.
- [40] PATEL-SCHNEIDER P.F. **A Revised Architecture for Semantic Web Reasoning.** *Lecture Notes in Computer Science*, vol. 3703 **2005**: pp. 32 – 36.
- [41] PATEL-SCHNEIDER P.F. AND FENSEL D. **Layering the Semantic Web: Problems and Directions.** In *Proceedings of The Semantic Web - ISWC 2002: First International Semantic Web Conference, Sardinia, Italy*, vol. 2342 / 2002. Springer-Verlag GmbH **2002**, p. 16.
- [42] POPESCU-ZELETIN R. **Implementing the ISO-OSI reference model.** In *SIGCOMM '83: Proceedings of the eighth symposium on Data communications.* ACM Press **1983**. ISBN 0-89791-113-X, pp. 56–66.
- [43] PRESSMAN R.S. *Software Engineering: A Practitioner's Approach.* sixth edition. McGraw-Hill **2005**. ISBN 0-07-285318-2.
- [44] SCHRAEFEL M.C., SMITH D.A., OWENS A., RUSSELL A., HARRIS C. AND WILSON M. **The evolving mSpace platform: leveraging the semantic web on the trail of the memex.** In *HYPERTEXT '05: Proceedings of the sixteenth ACM conference on Hypertext and hypermedia.* ACM Press, New York, NJ, USA **2005**. ISBN 1-59593-168-6, pp. 174–183. doi: <http://doi.acm.org/10.1145/1083356.1083391>.
- [45] SIMPSON H.R. **Layered Architecture(s): Principles and Practice in Concurrent and Distributed Systems.** In *1997 Workshop on Engineering of Computer-Based Systems (ECBS '97)* **1997**, pp. 312–320.
- [46] THURASINGHAM B. **Security Issues for the Semantic Web.** In *Proceedings of the 27th Annual International Computer Software and Applications Conference.* IEEE **2003**, p. 632.
- [47] USCHOLD M. **Where Are the Semantics in the Semantic Web?** *AI Magazine*, vol. 24(3) **2003**: pp. 25–36.
- [48] W3C. **Resource Description Framework (RDF) Model and Syntax Specification.** W3C Website **1999**. URL: <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>.
- [49] W3C. **XML Schema Part 0: Primer - W3C Recommendation.** W3C Website **2001**. URL: <http://www.w3.org/TR/xmlschema-0/>.
- [50] W3C. **XML Schema Part 1: Structures - W3C Recommendation.** W3C Website **2001**. URL: <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>.
- [51] W3C. **XML Schema Part 2: Datatypes - W3C Recommendation.** W3C Website **2001**. URL: <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>.
- [52] W3C. **Resource Description Framework (RDF): Concepts and Abstract Syntax.** W3C Website **2004**. URL: <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>.
- [53] W3C. **W3C Extensible Markup Language (XML).** Website **2005**. URL: <http://www.w3.org/XML/>.
- [54] ZIMMERMANN H. **OS1 Reference Model-The ISO Model of Architecture for Open Systems Interconnection.** *IEEE Transactions on Communications* **1980**.