The Future of the World Wide Web ?

Michael Wilson, Brian Matthews

W3C Office in the UK and Ireland
CCLRC Rutherford Appleton Laboratory,
Chilton, Didcot, OX11 0QX, UK
M.D.Wilson@rl.ac.uk, B.M.Matthews@rl.ac.uk
http://www.w3c.rl.ac.uk

Abstract. The Web started as a simple and very usable distributed system that was rapidly adopted. The Web protocols then passed through a period of rationalization and development to separate content from presentation in order to promote the re-usability of content on different devices. Today the developments in Web technologies are addressing new opportunities in Web Services and the Semantic Web, as well as the growing cultural diversity of the Web. These developments unite in the issue of *trust*, of content and services available on the Web, but also in access by others to the content and services that users may own. While the Web has been rationalizing, the Grid has developed to provide academic science with easier access to services and content. The Grid is now moving to exploit the robust interoperable commodity Web Services instead of maintaining its own middle level infrastructure. As Web Services, the Grid and the Semantic Web develop they will become increasingly interdependent on each other, and indistinguishable from the mainstream Web.

1 The Past

In 1991 Tim Berners-Lee, Robert Cailliau and Nicola Pellow from CERN released a portable line mode browser which could access documents held on distributed servers written in the HyperText Mark-up Language (HTML), through the HyperText Transport Protocol (HTTP), FTP or other protocols from a single address space within which each had a unique Universal Resource Location (URL).

The first major change to HTML came in 1993 when Marc Andreessen and Eric Bina from NCSA wrote the Mosaic browser and allowed in-line colour images through the introduction of the <IMG> tag.

A major addition to the overall Web architecture was also made in 1993 when Matthew Gray at MIT developed his World Wide Web Wanderer which was the first robot on the Web designed to count and index the Web servers. Initially, it only counted the available Web servers, but shortly after its introduction, it started to capture URLs as it went along. The database of captured URLs became the first Web database - the Wandex.

By 1993 these major components of the Web architecture were available from research organisations. The subsequent growth in Web pages and servers, and the development of commercial tools, which will not be recounted here. Such growth and commercial involvement show that the Web was a practical success that was becoming prey to competitive commercial interests. In 1994, as a result of concerns that fragmentation of Web standards would destroy the interoperability that the Web had achieved, Tim Berners-Lee and the Laboratory for Computer Science of MIT started the World Wide Web Consortium (W3C) to direct future developments for the Web.

The technologies which constitute the Web, although practical and successful, were technically crude. HTML combined the description of the content of a page with a description of the presentation of the page which limited the re-usability of the content, and included no type system to support static checking. There was no mechanism to present or compose time based media such as sound or video. The HTTP transport protocol was not optimized to the resource transfer usage of the Web. There was even confusion about what a URL could point to – files, but what of other resources such as devices (e.g. printers) and even people. However, these limitations allowed an accessible conceptual model and easy interaction without which it is unlikely that the Web would have been adopted.

W3C entered into a programme of work to reform these protocols, to overcome these problems, and incorporate technologies that would facilitate extensibility. To do this the simple architecture of URL, HTML and HTTP alone had to be sacrificed for a more complex one, where the new protocols and languages would no longer be easy to write in generic text editors, but would require specialized editors.

W3C addresses the core Web technologies that build on the transport layer standardized by the IETF and which are, in turn, built on by application specific standards that require a less rigorous process. Once a W3C working group is established to create a standard, or recommendation as W3C calls them, it usually takes two to three years before it is completed and officially published. Some groups have been terminated before completion when the motivation for standardization has dissipated or stakeholder consensus cannot be reached. The protocols published so far as recommendations by W3C are shown in Table 1.

2 The Present

The main four concerns of W3C today are listed below, while the architecture as it has evolved to meet them is shown in Figure 1.

Ensure access to the Web by many devices – The Web is becoming accessible from a wide range of devices including cellular phones, TV, digital cameras, and in-car computers. Interaction with resources on the Web can be achieved through a key pad, mouse, voice, stylus or other input devices. W3C has activities addressing device independence and multimodal interaction to contribute to W3C's goal of universal access.

Account for cultural diversity – To ensure access to the Web by people speaking different languages, with different writing conventions, and having different cultural backgrounds. In 1999, approximately half of all public Web sites were associated

with entities located in the United States, whereas by March 2003 only 35.2% of Web pages were in English, with 22.2% in Chinese or Japanese.
Web Services – the Web started as a tool for users to view documents. It is now moving to be a tool for computers to communicate with each other, on services provided by each other in a peer to peer manner rather than only in a browser to server one.

Table 1. Recommendations issued by W3C before March 2004

1996	1997	1998	1999	2000	2001	2002	2003	2004
PICS Rating	PICSRules	PICS DSig	Name spaces	DOM 2 Core	XHTML M12n	P3P 1.0	SVG 1.1	DOM 3 Validation
PICS Labels	HTML 3.2	CSS 2	CSS 1	ATAG 1.0	Canonical XML	XML Signature	DOM 2 HTML	CC/PP
		SMIL 1.0	WCAG 1.0	DOM 2 Events	Schema Primer	XML Canonicalization	SVG Mobile	Infoset (2nd)
		DOM 1	Style Sheets PI	DOM 2 Style	Schema Struct.	XHTML 1.0	XPTR Element	Namespaces 1.1
			MathML 1.01	DOM2Traversal	Schema Types	XPath Filter	XPTR Framewk	XML 1.0 (3rd)
			XPath 1.0	DOM 2Views	Ruby	Decrypt Transform	XPTR Xmlns	XML 1.1
			XSLT 1.0	XHTML Basic	XHTML 1.1	XML Encription	SOAP Adjuncts	OWL Guide
			HTML 4.01		XLink 1.0	UAAG1.0	SOAP Framewk	OWL Overview
					XML Base		SOAP Primer	OWL Reference
					SMIL 2.0		SOAP Tests	OWL Semantics
					SMIL Anim.		XForms 1.0	OWL Tests
					XSL 1.0		XML Events	OWL Use Cases
					WebCGM		MathML 2.0	RDF Concepts
							PNG (2nd)	RDF Primer
								RDF Schema
								RDF Semantics
								RDF Test Cases
								RDF/XML
								Speech Recognition
								VoiceXML 2.0

Semantic Web – as the data and service servers become more common on the Web, the data and services that they provide need to be described in a machine understandable way in order to be discovered, evaluated for fitness to a user's purpose and then called. The Web also needs to provide an environment in which contracts for the use of data and services can be established, and trust relationships defined to limit the growing incidence of cybercrime. To support machine dialogues in these areas, richer representations of ontologies, rules, and inference are required which are collectively termed the Semantic Web.

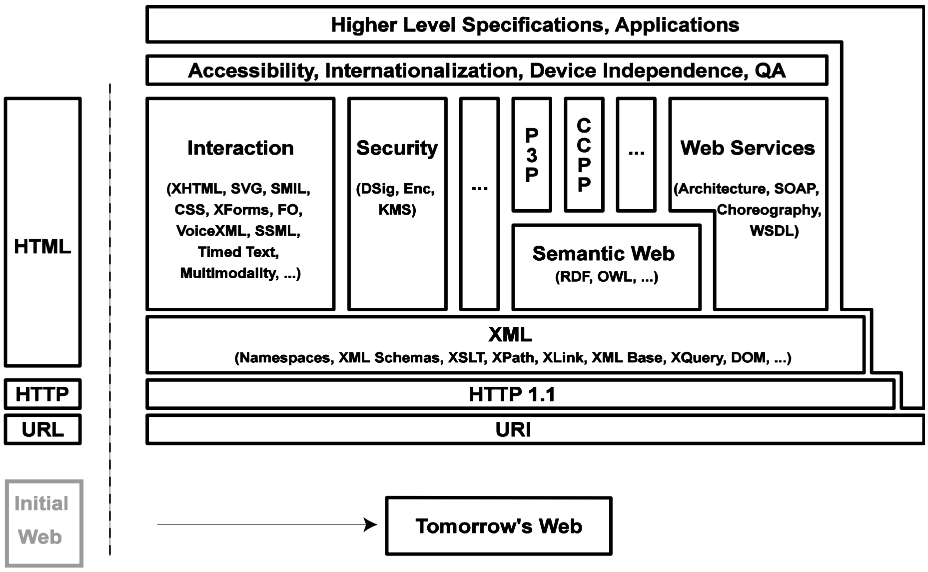

Fig. 1. The evolution of the architecture of the protocols in the World Wide Web

3 Web Services

Web Services have a long ancestry in distributed computing going back to remote procedure calls. The XML Protocol Activity which became the Web Services protocol or SOAP layer in the Web Services architecture was initiated in September 2000 in W3C following the observations that "distributed object oriented systems such as CORBA, DCOM and RMI exist with distinct functionality and distinct from the Web address space causes a certain tension, counter to the concept of a single space"[1]. As shown in Table 1, it was 2003 before any parts of SOAP reached the final recommendation form of publication.

In 2002 IBM and Microsoft agreed the main structure of the Web Services Architecture shown in Figure 2 (after [9]) which incorporated the Web Services Security component that is enlarged in Figure 3 (after [13]) since it has been so subdivided. These figures show the overall approach of a transport layer (HTTP) carrying messages (SOAP) between machines to invoke services. These services must describe their functionality (WSDL) and such descriptions can be registered in a directory (UDDI) to ease resource discovery. Privacy issues are still awaiting clarification , but most of the others have at least been proposed in pre-standardised forms. It took about 18 months for the various proposals for a Web Services Flow Language (WSFL) from IBM, Web Services Choreography Interface (WSCI) from BEA, Open Management Interface (OMI) from HP and WebMethods, to coalesce into BPEL4WS, and that is still not openly standardised, only supported by inter-company agreements. Similar time scales can be expected for any proposals for methods and interface languages higher up the stack as they pass through the development, standardization adoption and assimilation cycle.

One vision for motivating Web Services is that users will be able to use graphical business modeling tools to describe business models in terms of available services, with the desired quality and cost constraints on them. These services will be composed into the overall business that is being created in a language such as BPEL4WS, and meet various assurances and constraints on them which can be described in other languages. The business can then be created and operated through the tool.

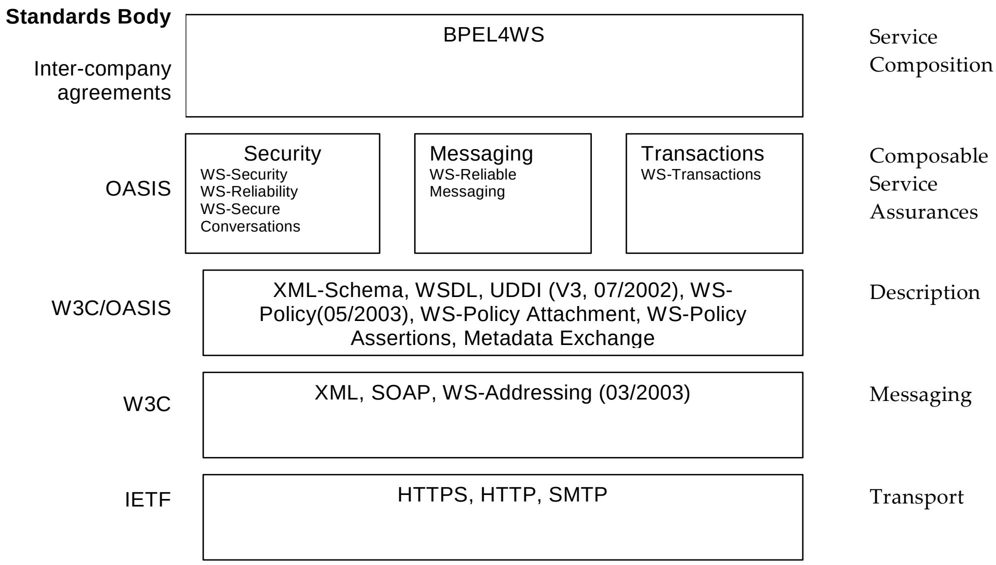

Fig. 2. The interoperable Web Services protocol architecture with the role of each layer shown on the right, and the standards body responsible on the left

The first generation of such Web Service composition tools are already available; for example, Triana from Cardiff University is an open source problem solving environment where the desired tools can be dragged to a work surface and wired together to create a workflow in BPEL4WS. It is too generic to yet include the specialised risk analysis tools and economic models to simulate whether prospective businesses would be profitable after composition, however it does support the basic composition and calling of Web Services through BPEL4WS. Such tools will become more specialized, and progressively incorporate resource discovery, trust and contract management components as they become available, adopted and assimilated.

Although there are commercial Business Process Management (BPM) tools available from many vendors, including IBM's CrossWorlds eBusiness Integration Suite, Microsoft's Biztalk Orchestrator and SAP's Business Workflow, there are still technical and interoperability problems with BPM over Web Services that many consider are not resolved in BPEL4WS alone. To address the service composition concerns W3C has recently initiated a Web Services Choreography Working Group with 32 participating organizations. The problems are exemplified by the number of alternative languages proposed by other consortia which do not relate cleanly to BPEL4WS, including XML Process Definition Language (XPDL) and Workflow-XML (Wf-XML) from the Workflow Management Coalition; Business Process Modelling Language (BPML) and Business Process Query Language (BPQL) from BMPI.

One of the general problems in service description and discovery is in the classic data and information retrieval problem of describing the thing to be discovered in a way compatible with the target's description. Along with the details of how to describe the quality of service and other constraints on the Web Service, this is addressed by the Semantic Web.

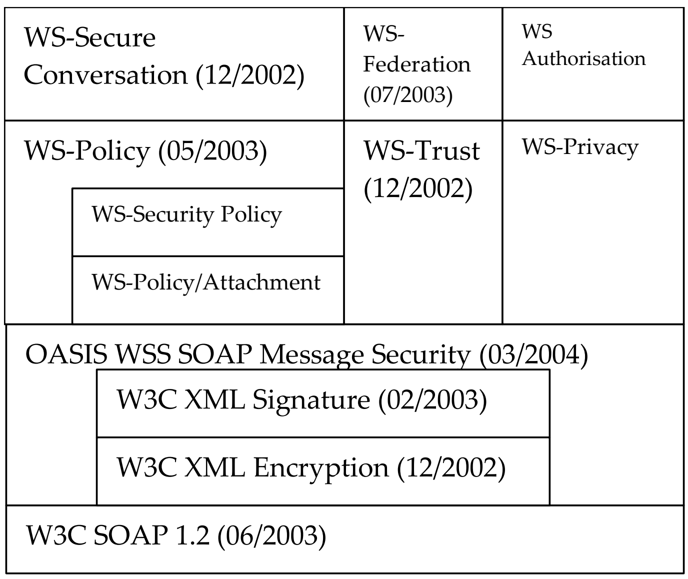

Fig. 3. Web Services Security Roadmap

4 The Semantic Web

One simple problem used to motivate the Semantic Web has been the need to discover resources on the Web, not only from their content as search engines do, but from descriptions of them. The problem is exemplified by the frustration in finding articles published by an author, rather than those which include the author's name. In response to the query "Time Berners-Lee" a search engine will respond with all the papers including that phrase, a subset of which will be authored by Tim Berners-Lee, but most of which will cite or refer to him – as this paper does. The Semantic Web should allow each resource on the Web to be described in metadata that states for data and information who its author was , when it was created, what its content it etc…, and for services what they do, what they cost and what quality measures apply to them. To do this it is necessary to define the structure of the metadata and a restricted, and machine interpretable, vocabulary in which to describe it. In turn, it is then necessary to define a language to express these two. The language initially proposed by W3C was the Resource Description Framework (RDF). A common metadata format for Web resources was also proposed in the form of the Dublin Core [7] which defines 15 elements of metadata, so that both the Dublin Core elements and the content can be stated in RDF.

Unfortunately the solution is not that simple. The use of RDF and Dublin Core for resource discovery is analogous to the use of HTML and HTTP for the Web in 1993. It is superficially attractive, and appears to solve the immediate resource discovery problem, but it does not address all the issues that the specialists in knowledge acquisition and ontology management, maintenance and reasoning raise about simple lists of words. A more elaborate architecture is required to address these problems that immediately arise as is shown in Figure 4 – after [3].

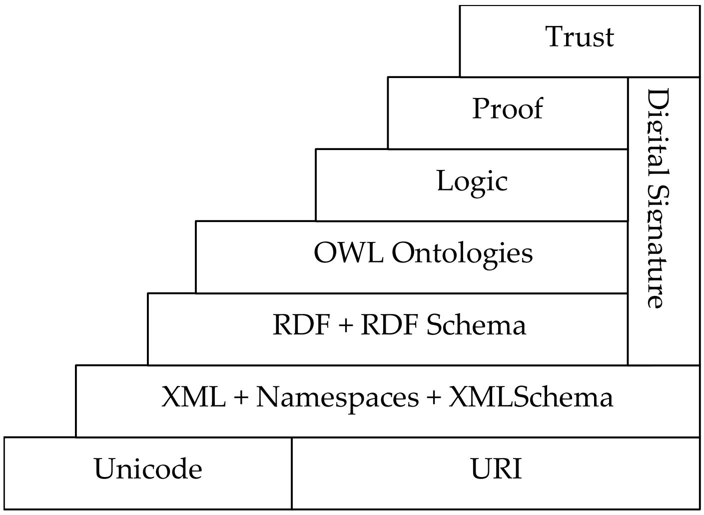

Fig. 4. The Semantic Web Architecture.

When RDF and Dublin Core alone are used for the creation of metadata an one's own Web pages the two main problems are with the creation, maintenance and interoperability of the vocabulary used, and with the trustworthiness of the metadata. The vocabulary cannot be reasoned over or proven to be complete or consistent to any rules. If the vocabulary is stated in an Ontology language such as OWL, then tools can apply Description Logic [6] to test its internal consistency and identify combinations of properties for which no terms exist. The second problem is harder to overcome - why should a resource discovery tool trust the metadata a Web page author states on their own page since many sites will place every possible term to attract the most potential readers or users of the service? The lack of accessible vocabularies and the trust problem have prevented the take up of metadata for resource description so far.

RDF and RDF Schema have been released as recommendations by W3C this year, and several robust tool sets exist for using them which should foster their use. For example the Jena toolkit from Hewlett Packard supports the creation and query of RDF knowledge bases, as well as several different inference engines that can reason over the knowledge. Equally work is underway to provide easy migration from existing vocabulary resources such as thesauri into RDF so that they can be supported by generic Semantic Web tools. RDF query which is supported by such tools has just started its path through the standardization process at W3C.

Similarly, OWL has become a recommendation this year, with the result that various vocabularies that have been created in its precursors can now be used in a standard representation, which should motivate the development of robust OWL based tools for ontology creation and management. One example is the DAML-S vocabulary to define quality measures on Web Services [16]. The standardization of an agreed OWL-S based on this is being considered by W3C at present. This would allow a standard language to be used by Web Service composition tools for properties of Web Services.

RDF and OWL are each languages that fulfill specific purposes in resource and service description. However, RDF is also a generic language for defining vocabularies and graphs beyond hierarchies, which XML describes. Therefore it can be used for many other applications, and the tools for it become generic as those for XML have done. There are many benefits in this for the users of the languages who only need to learn the generic language and tools for a wide range of applications rather than a new language for each one. This benefit is consistent with the approach taken to the Semantic Web development, that investment in each layer of the pyramid should provide a profitable return at that layer alone, so that a single massive investment is not required in all the technologies at once. For example, RDF itself is used to support applications such as the news syndication language RSS, a person networking language FOAF, as well as applications in calendar management. Each of these is addressed by other applications, but the use of a generic technology and tool set allows the interaction of the knowledge stored in each application to provide, for example, the graphical display (through SVG) of the geographical location and timeline representation of activities from the FOAF and calendar data.

Further up the pyramid are the logic and proof layers where there are several proposals for rule languages based on RDF such as RuleML [15] and N3 [2]. These are developing demonstration applications to illustrate and test their structure and tool support, but they have not yet been widely adopted, nor do they yet approach standardization. One of the problems with this layer is that it raises the specter of artificial intelligence which many business investors consider either to be too long term an investment, or have memories of losing their investment in the 1980's expert system boom which they do not wish to repeat. One important issue to be overcome before these fears can be allayed is that of the liability for actions taken by rules in distributed systems. There is concern that until distributed rule based systems are sufficiently established and contracts applying to them are standard practice based upon accepted legal precedent, then anybody in the development and use chains could be liable for compensation claims following errors.

The final layer in the pyramid addresses these issues of trust and the legal contracts that apply across the Semantic Web and Web Services.

5 Trust and Contracts

Trust and legal contract management are issues where Web Services and the Semantic Web meet, and are only now being addressed within this integrated approach.

Trust between parties (e.g. buyer and seller) in turn introduces a wider context of the contract or relationship in which the trust exists. The trust will apply to an entity in a contract to fulfill a role. To extend the Web Services architecture upwards, trust policies can be imposed on the business process management layer to constrain the lower security layer in the light of a contract, which itself is established to implement a business model in the context of the prevailing law of the country or international situation. This is illustrated in Figure 5. However, given what was said above about the current standardization and interoperability of business process management, clearly any edifice built upon it is unstable at present and still a research topic.

Fig. 5. The Trust Stack.

Trust judgments can be made on individuals or organizations (trustees) using Semantic Web technologies to model and reason about them. The basis of trust judgments are the trustor's experience of the trustee and any available recommendations from others as to their trustworthiness; which may themselves be based on experience. There is a body of work on the process or collecting evidence for, and making such judgments using Subjective Logic based on the established Dempster Shaffer evidence theory [14] which is being incorporated into the Semantic Web technologies. There are also demonstrator systems for managing access control to information, data, and computational resources which include trust based authorization decisions (e.g. [12]) that can apply over the roles represented in the business process management (BPM) layer to propagate constraints to the security layer to authorize access to data and use of services and other resources. This research is also starting to introduce the delegation of obligations resulting from contracts, although this is still at a very early stage, and does not yet include the refinement of responsibilities to action plans during delegation between roles in the BPM, although it does address the concomitant delegation of authority.

The most significant recent development at the contract layer itself is the Web Services Service Level Agreement language – WSLA [5]. This requires a range of vocabularies used in WSDL that should be maintained with Semantic Web technologies. It applies them to Web Services, but is not an integral part of the Web services protocol architecture, fitting more easily into the higher level trust stack since it applies a clear legal contract to each Web Service transaction.

The WSLA language elements include those that normally occur in contracts to define liability and binding complex terms and conditions for operations with their quality of service and computational assurances as well as price :

Parties	e.g. Service provider, consumer
Action Interfaces	Interface descriptions
Sponsors	e.g. Measurement Service
Service Description	Common view of the service
Service Objects, refer to	Specification, link to serv. descr.
- WSDL	e.g. WSDL service, port,
- BPEL, …	binding, operation
SLA Parameters	e.g. Response time
Metrics	e.g. Transaction counter
Measurement Directives	e.g. Sampling interval
Functions	e.g. (metric1 + metric2) /2
Obligations	What is actually promised
SLOs, ActionGuarantees	e.g. Notify management service

WSLA is the first language to capture the legal contractual terms from a mainstream supplier. It will probably not survive in its present form to widespread adoption but given the timescales for standardization and adoption mentioned previously, it would be reasonable to expect several alternative proposals over the next two years when a standardization process could start, completing in about five years.

6 The Web and the Grid

While the Web has been developing, the demands of scientists for sharing supercomputers, using the spare compute cycles on the many desktop machines in universities, and passing vast amounts of experimental data have lead to the development of the Grid [10]. The fundamental difference between the Grid and the Web has been that each transaction in the Web is discrete, in that the connection between client and server is broken and the only way to maintain the state of the dialogue is to deposit a copy of it (e.g. as a cookie) with the client. In contrast the Grid calls services and maintains the state of the dialogue so that the progress being made to the objective of the service can be monitored (e.g. report on how far a computation has progressed). Recent design decisions in the GGF who standardize the GRID, have been to move to an Open Grid Services Architecture (OGSA) based on Web Services, but including the ability to model stateful resources- WS-Resource Framework (WSRF). WS-Resource [11] has been introduced recently to support the declaration and implementation of the association between a Web Service and one or more named, typed state components. WS-Resource models state as stateful resources and codifies the relationship between Web Services and stateful resources in terms of the implied resource pattern - a set of conventions on Web Services technologies, in particular WS-Addressing. When a stateful resource participates in the implied resource pattern, it is referred to as a WS-Resource. A WS-Resource is defined and associated with the description of a Web service interface while its properties are made accessible through the Web Service interface that can be managed throughout the WS-Resource's lifetime. With the introduction of OGSA and WSRF the advances made for science in Grid computing can be made available on top of the generic commodity Web Services, while freeing up the scientific development community to focus on their needs instead of developing the security, trust and semantic components for the Grid independently, since they are available at the underlying levels. As the Semantic Web and Web Services interact and are moving to a single Web, so the Grid is moving to be a layer on top of Web Services rather than a separate distributed technology.

7 Conclusion

Although the Web is a successful commodity that has millions of active users, it is not set in an unchanging form. Advances are expected in Web Services, the Semantic Web and the integration of the Web and the Grid. These should provide a basis for the creation of businesses from Web Services drawing on the vocabularies available through the Semantic Web, suitable to meet the legal requirements of national and international law. The encoding, and enforcement of legal contracts has only just begun, but it will be an active area of research with a clear financial benefit to many parties which also requires advances in many areas of computer science. More advanced topics such as the agent based negotiation of contracts with trustworthy, or untrustworthy, actors that are dependent on these (e.g. [8]) remain still further along the Web Roadmap (see [4]).

References

Details of the W3C recommendations mentioned in the text are available online at http://www.w3.org

1. Berners-Lee, T (1998) Web Architecture from 50,000 feet, http://www.w3.org/DesignIssues/Architecture
2. Berners-Lee, T (1998) Notation 3: An RDF Language for the Semantic Web http://www.w3.org/DesignIssues/Notation3.html
3. Berners-Lee, T., Hendler, J., Lassila, O., The Semantic Web, Scientific American, May 2001
4. Berners-Lee, T and Connelly, D. (2001) W3C Roadmap Diagrams – http://www.w3.org/2001/04/roadmap/
5. Dan, A. et. al., "Web Services on Demand: WSLA-Driven automated Management, IBM Systems Journal, Vol 43, No 1, 2004
6. S. Decker, D. Fensel, F. van Harmelen, I. Horrocks, S. Melnik, M. Klein, and J. Broekstra. Knowledge representation on the web. In Proc. of the 2000 Description Logic Workshop (DL 2000), pages 89-98, 2000.
7. DublinCore (1995) The Dublin Core Metadata Initiative - http://dublincore.org/
8. Fatima, S.S, Wooldridge, M and Jennings M.R. (2003) An Agenda based framework for multi-issue negotiation. Artificial Intelligence.
9. Ferguson, D.F., Storey, A., Lovering, B., Schewchuk, J., "Secure Reliable, Transacted Web Service: Architecture and Composition" (2003), http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwebsrv/html/wsoverview.asp
10. Foster, I., Kesselman, C., (1997) Globus: A Metacomputing Infrastructure Toolkit. Intl J. Supercomputer Applications, 11(2):115-128.
11. Global Grid Forum, The WS-Resource Framework (March 2004) http://www-106.ibm.com/developerworks/library/ws-resource/ws-wsrf.pdf
12. Grandison, T. and Sloman, M. SULTAN -- A Language for Trust Specification and Analysis Proceedings of the 8th Annual Workshop HP OpenView University Association (HP-OVUA) Berlin, Germany June 24-27 2001 http://www.hpovua.org/PUBLICATIONS/PROCEEDINGS/8_HPOVUAWS/Papers/Paper01.2-Grandison-Sultan.pdf
13. IBM and Microsoft, 2002 Security in a Web Services World: A Proposed Architecture and Roadmap – http://www-106.ibm.com/developerworks/webservices/library/ws-secmap/
14. Josang. An algebra for assessing trust in certification chains. In J. Kochmar, editor, Proceedings of the Network and Distributed Systems Security (NDSS'99) Symposium. The Internet Society, 1999.
15. Michael Schroeder and Gerd Wagner (Eds.): Proceedings of the International Workshop on Rule Markup Languages for Business Rules on the Semantic Web. Sardinia, Italy, June 14, 2002. CEUR-WS Publication Vol-60. - http://www.ruleml.org
16. OWL-S Coalition (2003) OWL-S 1.0 Release http://www.daml.org/services/owl-s/1.0/