Ubiquitous Web Services

Malcolm Attard

Department of Computer Science and AI,
University of Malta

1 Introduction

Ubiquitous coming from the Latin word *ubique*, means existing or being everywhere, especially at the same time. Web Services are loosely specified and coupled components distributed over the internet [23] with the purpose of being accessed and used ubiquitously by suppliers, customers, business and trading partners. This must be done independently of any tools or environment in use by any party involved. The basic service oriented architecture is based on the publishing of a service by a service provider, the location of a service by a service requestor and the interaction between the two based on the service description. The necessary functionality for the full adoption of such web services must include routing, reliable messaging, security, transactions, binary attachments, workflow, negotiation and management, web services description languages, choreography, orchestration and non-repudiation. A large number of companies and organizations are promoting this adoption and shifting their strategy to include this useful technology. A multitude of proposed standards and products have emerged in an attempt to meet the needs of this worldwide community of web services adopters. The core established standards include the *Web Services Description Language* (WSDL), *the Simple Object Access Protocol* (SOAP) and the *Universal Description, Discovery and Integration* (UDDI). The *Web services Inspection Language* (WSIL) is a more lightweight yet complimentary specification for service discovery[1]. Other definitions produced to tackle the required functions have not been fully standardized and many are still competing. For the needed functionality to be produced a number of related issues must be tackled. Here we look at some of the important ones, and how they are being tackled, we then shortly describe our proposed project and related works.

2 Transactions

Transactions are essential factor for web services adoption. In traditional scenarios the standard properties for a transaction are atomicity, consistency, isolation and durability (ACID). Now since most web services based applications are distributed remotely and usually owned by different parties, normal methods for transaction management are not as effective. Having central control over all the resources is very difficult in this scenario. The *Business Transaction Protocol* (BTP) specification aims to solve these problems by extending conventional methods to enable both ACID and non-ACID transactions using a two phase commit model based on structures defined as ATOMS and Cohesions [11]. The *WS-Transactions and WS-Coordination* specifications, recently released from WS-I, are specifications which like BTP aim to provide a mechanism where systems can inter-operate transactionally. The WS-Coordination framework defines services which include:

- An activation service
- A registration service
- A coordination service

while the WS-Transactions specification defines two type of protocols:

- atomic transactions protocols for short lived transactions
- business transactions protocols for long lived transactions

They provide the mechanism where transactions are described as series of activities. Services are created and registered via WS-Coordination services and their execution coordinated by the WS-Transaction protocols [5].

3 Security

A "user" may have a number of identities which need to be handled across different systems and services. The optimal approach would be to have a universal log-on, but this is currently impractical. Currently the leading initiatives in Web Services Security are the SAML, the Liberty Alliance Project based on SAML and WS-Security from WS-I [9]. SAML's objective was to enable interoperability of security services across the internet, where security information is transmitted using XML. SAML provides a way to encapsulate the authentication process and provide transport for it. Thus the authority can determine what authentication to use. As discussed in [10] Microsoft have already attempted to use the Microsoft .NET Passport for universal single sign-on mechanism. Now *WS-Security* which is an XML and SOAP based message security model from IBM, Microsoft and Verisign has been recently submitted to OASIS. WS-Security extends the use of XML Encryption and XML Signature for protection and verification respectively. It has a higher level of abstraction than SAML and thus enables it to include SAML as a supported technology. Like SAML it does not specify authentication mechanisms but uses SOAP messages and describes how to attach signature and encryption headers to SOAP. The *Liberty Alliance Project* is focused towards a federated authentication framework where multiple identities can be linked together with the user's consent. Liberty is based on three specifications:

- XML driven ways to communicate authentication information
- How these map to HTTP, SOAP and mobile protocols
- The authentication context

SAML is used as the assertion language for Liberty.

4 Quality of Service

Quality of service is another important factor for consideration when using web services. The following are major requirements [13] for a quality web service

- Availability - service needs to be present and ready for use.
- Integrity - maintain correctness of interaction.
- Accessibility - be capable of serving a web service request.
- Performance - have a certain level of throughput and latency.
- Reliability - maintain the service itself and the service quality.
- Regulatory - comply with standards and conform to the devised rules.
- Security - provide the agreed level of security.
- Transactional Qualities - conserve the stabilized transactional behavior.

WSDL does not specify semantics or aspects regarding the Quality of the Service. Thus QoS must be described in some other way. A lot of work has been done on QoS web based services at different levels including the network level, the system level, the web server level and the service level[22]. A service may be deployed to provide different service levels and assurances to different clients. Negotiation of web services and their QoS properties usually involves the creation of Service Level Agreements (SLAs). These must then by enforced, monitored and when expired, terminated as well. Third parties may also be involved in monitoring the QoS of a particular service [4].

5 Semantics

The Semantic Web is an extension of the current web in which information is given well-defined meaning [2]. It is based on RDF standards and driven by the W3C, together with a large movement of researchers and industrial partners. Its objective is have data defined and linked in such a way to achieve better use of the information on the internet. The DARPA Agent Mark-up Language has the purpose of marking up web pages such that they are given meaning, such that a DAML enabled browser or search engine may produce a better result than currently syntactically based ones. The DAML group of languages includes DAML-ONT,OIL, DAML+OIL and DAML-L. DAML+OIL[12] was produced from the convergence of the DAML-ONT, the first ontology language and OIL (Ontology Inference Layer) a logic based system. DAML-L is on the other hand is a complementary logical language which is able to express at least propositional Horn Clauses. These languages can be used to define the terms needed for the description of service invocation [8]. DAML-S is a proposed DAML+OIL ontology which has the purpose of describing the behavior, properties and capabilities of web services [15].

6 Composition

Web Services Composition involves the combination of a number of web services to produce a more complex and useful service. Choreography is the term used to define the tracking of message exchange between services while the term orchestration is used to refer to the services interaction involving the logic and order of interaction execution[17]. A number of composition languages have emerged to meet this purpose. These include IBM's Web Services Flow Language (WSFL) and Microsoft's XLANG whose concepts have been placed in the Business Process Execution Language for Web Services (BPEL4WS), a new specification from WS-I whose core members intuitively include Microsoft and IBM. This specification describes the modelling of web services behavior using business process interaction. The Web Services Choreography Interface (WSCI) is another specification produced by BEA, SAP and Intalio. It describes messages between the web services such that it defines the choreography as a exchange of messages. BPML produced by BPMI.org is yet another mark up language which takes the form of a meta language for describing business processes[25]. Using these representations and the relative composition engine one can compose web services as desired.

7 UbiWSCo

As the number of web services increase drastically with time and their use is extended to more common services, the task of effectively composing a web service will become overwhelming.

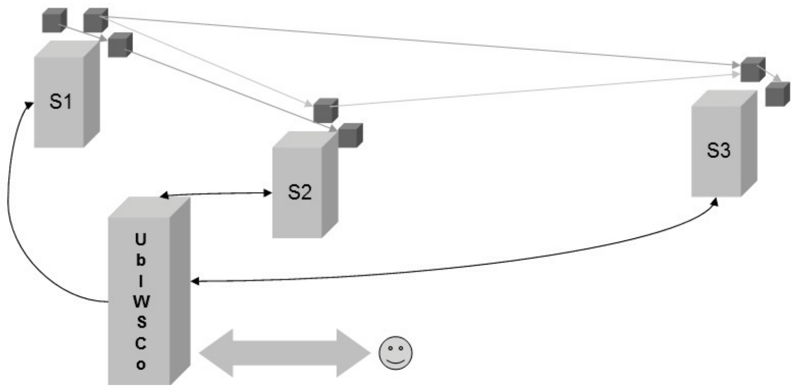

Fig. 1. Service Composition across multiple servers

We aim to provide a detailed study of current web service composition definitions and methodologies and the development of a method, accompanied by its implementation (Ubiquitous Web Services Composer), which will enable us to compose a new service, from a number of other web services, automatically. This service will base its composition strategy on statistically based methods and user modelling where 'end-user' refers to either a company software business system, agent, or human user. Our approach will take into consideration pre-defined compositions, service usage history, the actual user needs, the services' SLAs and also the usage context of the service. The 'end-user' will be offered a search like user interface which will upon request provide a number of plausible compositions which can then be deployed by the 'end-user'. The selection criteria which will be used will also include transactional qualities and access options. We intend to use BPEL4WS together with the complementary specifications WS-Transactions and WS-Coordination based definitions for the purpose and to construct our system using Java driven IBM alphaworks technologies. We also intend to examine the application of such a mechanism and its interrelation with user facing web services [19].

8 Related Work

Web Services composition is a an active area of research and there exist a number of related frameworks which we will shortly outline here. The *WeSCoS* (Web Service Composition Framework) [14] was developed to provide a research framework for composition of loosely coupled services. It uses the XML programming language (XPL) for invoking services, iterating over results and creating output information in a document. In the *ICARIS project* [24] is based on Jini, JavaBeans and XML. It consists of two core management components, the Registration Manager and the Composition Manager, working on the Jini Infrastructure itself. The Composition Manager is responsible for the actual composition of general services while the Registration Manager is responsible for managing registration access rights. Composition is achieved using JavaBeans and the ERCSP (Extensible Runtime Containment and Services Protocol). The framework is generic and is not based on Web Services. Another automatic composition approach described in [26] is based on the semantic matching of web services parameters. *METEOR-S*, also from LSDIS is a follow up project to METEOR (Managing end-to-end operations) and provides MWSCF: (METEOR-S Web Service Composition Framework), a comprehensive framework for semantic web service composition. Within this framework the existing process composition techniques are enhanced by using templates to capture the semantic requirements of the process. In [16] another pattern composition approach is described. It involves the use of DAML-S subset, a situation calculus as first order logic language for the description of changes based on named actions and petri nets for execution semantics. We also see the use of a petri net-based algebra to model control flows for web services composition in [7]. The *QUEST* framework [6] which is able to provide the best initial service composition path chosen using multiple QoS constraints and a dynamic re-composition mechanism to provide service path change on the fly, also based on QoS constraints and violations. *SWORD* is another tool set for Web Services Composition which uses a rule-based expert system to determine whether a service can be composed using existing services [18]. SWORD is focused towards information providing services and it can generate a functional composition plan given the functional requirements. HP eFlow supports specification, deployment and management of composite web services. A composite service in eFlow is described as a schema and modelled by a graph including service, decision and event nodes. To achieve adaptive service processes it provides a dynamic service discovery mechanism, multi-service nodes (allowing for multiple parallel activation of the same service node) and generic nodes (non statically bound services) [3]. *Self-Serv* is a framework based on state-chart modelling techniques where transitions are labelled using ECA rules. Thus services are composed in a declarative manner and the orchestration engine is based on a peer to peer model such that processing is distributed among the participants of the orchestration [21, 20].

References

1. Timothy Appnel. An introduction to wsil. The O'Reilly Network, October 2002.
2. Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, May 2001.
3. Fabio Casati, Ski Ilnicki, LiJie Jin, Vasudev Krishnamoorthy, and Ming-Chien Shan. Adaptive and dynamic service composition in eflow. Technical report, Hewlett-Packard Co., 2000.
4. Asit Dan, Heiko Ludwig, and Giovanni Pacifici. Web service differentiation with service level agreements. *IBM developerWorks*, May 2003.
5. Tom Freund and Tony Storey. Transactions in the world of web services, part 1. *IBM developerWorks*, August 2002.
6. Xiaohui Gu, Klara Nhrstedt, Rong N. Chang, and Christopher Ward. Qos-assured service composition in managed service overlay networks. In *23rd International Conference on Distributed Computing Systems*, May 2003.
7. Rachid Hamadi and Boualem Benatallah. A petri net-based model for web service composition. In *Fourteenth Australasian Database Conference (ADC2003)*, volume 17 of *Conferences In Research and Practice in Information Technology*. Australian Computer Society, Inc., 2003.
8. James Hendler. Agents and the semantic web. *IEEE Intelligent Systems*, March-April 2001.
9. Stuart J. Johnston. Positive identification. *XML and Web Services Magazine*, October-November 2002.
10. Stuart J. Johnston and Dan Ruby. Federated indentity face-off. *XML and Web Services Magazine*, October-November 2002.
11. Muhammad F. Kaleem. Transactions over web services - an introduction to the business transaction protocol, May 2002.
12. Deborah L. McGuinness, Richard Fikes, James Hendler, and Lynn Andrea Stein. Daml+oil: An ontology language for the semantic web. *IEEE Intelligent Systems*, 17(5), Semptember-October 2002.
13. Anbazhagan Mani and Arun Nagarajan. Understanding quality of service for web services. *IBM developerWorks*, January 2002.
14. Laurence Melloul, Trevor Pering, and Armando Fox. Wescos: A first step towards web programming. Technical report, Stanford University.
15. S. McIlraith, T. Son, and H. Zeng. Semantic web services. *IEEE Intelligent Systems*, March-April 2001.
16. Srini Narayanan and Sheila A. McIlraith. Simulation, verification and automated composition of web services. In *Eleventh International World Wide Web Conference*, 2002.
17. Chris Peltz. A review of emerging technologies, tools and standards. Technical report, Hewlett Packard, Co., January 2003.
18. Shankar R. Ponnekanti and Armando Fox. Sword:a developer toolkit for web service composition. In *The Eleventh World Wide Web Conference (WWW2002)*, 2002.

19. Eilon Reshef. Building interactive web services with wsia and wsrp. *Web Services Journal*, 2, December 2002.
20. Quan Z. Sheng, Boualem Benatallah, and Marlon Dumas. Self-serv environment for web services composition. *IEEE Internet Computing*, pages 40–48, January-February 2003.
21. Quan Z. Sheng, Boualem Benatallah, Marlon Dumas, and Eileen Oi-Yan Mak. Self-serv: A platform for rapid composition of web services in a peer-to-peer environment. In *Proceedings of the 28th VLDB Conference*, 2002.
22. Akhil Sahai, Jinsong Ouyang, Vijay Machiraju, and Klaus Wurster. Specifying and guaranteeing quality of service for web services through real time measurement and adaptive control. Technical report, E-Service Management Project, E-Services Software Research Department, HP Laborataries, 1501 Page Mill Road, Palo Alto, CA 94034, 2001.
23. Brent Sleeper and Bill Robins. Defining web services, 2001.
24. Vladimir Tosic, David Mennie, and Bernard Pagurek. Dynamic service composition and its applicability to e-business software systems - the icaris experience, 2000.
25. W.M.P. van der Aalst. Web services – been there done that? *IEEE Intelligent Systems*, January-February 2003.
26. Ruoyan Zhang, I. Budak Arpinar, and Boanerges Aleman-Meza. Automatic composition of semantic web services. In *Intl. Conf. on Web Services, Las Vegas NV*, 2003.