# A FRAMEWORK FOR WEB SERVICES SECURITY POLICY NEGOTIATION

**Tristan Lavarack[1] and Marijke Coetzee[2]**

Academy for Information Technology

University of Johannesburg

South Africa

[1]200506397@student.uj.ac.za

[2]marijkec@uj.ac.za

ABSTRACT

In today's business environment, the use of web services technology is becoming more popular. This growth has been met with an increase of security related attacks, which has caused web services providers to adopt stricter security policies. As not all web service consumers can implement the security requirements of web services providers, they may turn to use the services of other providers. In order to address this problem, this paper introduces a framework for a web services security policy negotiation system that web services consumers and providers can use to negotiate a customised security contract. The framework is defined over current web services technology, to be used by business-to-business (B2B) web services collaborations. The inflexibility of current security policy specification languages for negotiation is overcome, by incorporating human intuitiveness supported by an intelligent negotiation support system.

KEY WORDS

Web Service, NSS, Negotiation, Security, Security Policy

# A FRAMEWORK FOR WEB SERVICES SECURITY POLICY NEGOTIATION

## 1 INTRODUCTION

A web service is an autonomous, well-defined, standards-based component that is accessed via web-based protocols. Such services enable the dynamic assembly of B2B (business-to-business) functionality, across loosely coupled heterogeneous platforms. The increased usage of web services technology has lead to a growing number of malicious attacks in this environment [1]. This aspect, coupled with the importance of web services as an integration technology, has led to concerns over their security [2].

To protect themselves from attack, web services providers define security policies that describe their capabilities and requirements such as identification and data integrity. Web services providers may expect of all web services consumers to adhere to these security policies to be able to use the web service's functionality. If web services consumers cannot apply all stipulated security policy requirements, they will have to search for alternative web services providers. In such as case, a web services provider loses these consumers, and may rather decide to negotiate some aspects of its security policy requirements so that interaction can take place. This paper highlights important requirements for a security policy negotiation system, designed to fit into the current web services architecture [3].

The remainder of this paper is organized as follows. The next section provides a background on web services and web services security. Section 3 discusses related work on negotiation. Section 4 examines a list of requirements for a web services negotiation system. Section 5 provides a framework for the web services negotiation system. The paper ends with a conclusion and a brief look at future work.

## 2 BACKGROUND

The web services architecture [4] identifies web services providers and web services consumers as two endpoints of communication. A web services provider can be seen as the machine that holds the web services implementation, which the web services consumer utilises. Web services support interoperable machine-to-machine interaction over a network [4]. They have an interface described in WSDL [5] and consumers interact with the web service via SOAP messages. WSDL represents a valuable tool for the description of functionality but not for security capabilities and requirements. Therefore, secure interoperability requires additional mechanisms. Security capabilities and requirements of a web services provider are stored in policy documents. Such documents are attached to specific services and are made available to web service consumers, to allow them to consume the web service successfully. There are many different policy languages that can be used to define security policies such as WS-Policy [6], WS-Security Specification [7], WSPL [8] and WS-Agreement [9]. For semantic web services, KAoS [10], [11] can be used.

Web services that are used in business-to-business (B2B) transactions have high requirements for security. Consequently, it is more complex for large numbers of web service consumers, each having his/her own set of security requirements, to apply to the full set of security requirements of a web services provider. The problem can be described as follows: a web service provider's security policy requires that consumers perform authentication with certificates, encryption with AES, data integrity with MD5, and roles for access control. If a web service consumer uses username-password combinations for authentication, and does not support AES or roles, the web service consumer will either have to change its security mechanisms, or search for an alternative web services provider. To solve this, web services security policy negotiation can be used to negotiate over these security requirements, by taking into account the needs and limitations of each party. The result of the policy negotiation is a security contract where encryption with DES, data integrity with MD5, and roles for access control are agreed to, but a compromise is made to use username-password combinations for authentication, as the consumer has a good reputation.

The security contract is thus an agreed upon set of security clauses between the web services provider and web services consumer, whereas a security policy is a set of security requirements of a single party. Such a security contract can be used to monitor the interactions between the web service and the web service consumer, to enable service governance. Even though it may be more costly, the ability to customise web services security contracts gives web services providers more flexibility to be able to attract potential consumers.

In the next section, recent research on web services security policy negotiation is discussed.

## 3 RELATED WORK

There is a large body of research on negotiation [12], [13], [14], [15], [16]. Negotiation can be defined as a decentralised decision-making process by at least two parties. It is performed until an agreement is reached, or the process is terminated without reaching an agreement [17], [18]. For web services, related research in this field has addressed the negotiation of Service-Level Agreements (SLA) in the Grid environment [19], WS-Negotiation [18] focusing on negotiation of business parameters, and negotiation of privacy policies [20].

The negotiation of policies such as security and privacy has been identified as an important research focus for web services [21]. Security policy negotiation refers to the adjustment in security requirements and capabilities, to accommodate needs of both consumers and providers and their environmental conditions. To date, no industry solution has been defined for this difficult problem, with little research addressing negotiation using current web services standards such as WS-Policy [6]. A prototype developed by Korba and Yee [1] is representative of the most resent research on this topic, and is discussed next.

The prototype implements a semi-automated security policy negotiation system, based on a peer-to-peer architecture. It enables an Internet service, which does not necessarily have to be a web service, and its clients, who may be human, to contact each other and hold a negotiation session across the Internet. Both the Internet service and the client have their security requirements in a security policy document. The

negotiation system evaluates both the Internet service's and the client's security policies. Where the security requirements are the same, a match is made and no changes are necessary. Where the security requirements do not match, negotiation has to be performed. As part of the semi-automated approach, the administrator of the Internet service and the owner of the client are given the opportunity to edit their security policies to create a new security policy in order to accommodate each other. Policies are exchanged, re-evaluated, and the process continues until an agreement is reached or until either the Internet service or client terminates the negotiation process. As it is difficult to evaluate security policies, either party can ask the system for help. The help module provides a human located at each party, with a history of past choices that have been made by many others in similar situations. This is done to help them understand which security requirements were preferred before, and how many times each has been used. The help system thus provides "best practice" to its users.

Limitations of the prototype are that it is not specifically designed for B2B web services interactions; the help module does not provide any information on how decisions affect each other, policy decisions are inflexible as a match is made or not, and the prototype does not consider the role that the state of the environment, and type of relationship between parties, play in the negotiation process.

Next, the requirements for a web service security policy negotiation system, with the aim of extending this prototype are discussed.

## 4 WEB SERVICES SECURITY POLICY NEGOTIATION REQUIREMENTS

The ability to negotiate security contracts is an important feature to be addressed for secure and flexible B2B web services interactions. Requirements for such a system need to consider the type of negotiation strategy, as a fully automated process may not be practical since security is a high risk. There is a need for an intelligent support system for administrators to assist them with decision-making. To be able to define such a system, the following seven requirements have been identified by this research:

1. Standards-based implementation
2. Standards-based security policy specification language
3. Standards-based negotiation protocol
4. Semi-automated negotiation strategy
5. Negotiation support system
6. Collaborative decision-making
7. Consideration of environmental influences

These requirements are next discussed in greater detail.

### 4.1 Standards-based implementation

B2B web services interactions, in which business relationships change regularly require a highly flexible security framework based on approval and universal acceptance of standards. This allows business partners to avoid interoperability problems among their disparate information systems. The adoption of web services standards is thus important. As no current standard directly addresses policy negotiation, a solution needs to be found to cater for this need without affecting interoperability. The requirement for a standards-based implementation influences choices to be made with respect to the policy specification language, negotiation protocol, negotiation strategy and decision-making, discussed next.

### 4.2 Standards-based security policy specification language

The choice of the policy specification language has a far-reaching effect on the web services policy negotiation system, and needs to be carefully considered. A major consideration is that security policies should be based on standard technologies, so that runtime platforms can read, interpret and enforce the security policy.

As mentioned, there are a number of languages that can be used such as WS-Policy, WS-SecurityPolicy, XACML, WSPL and WS-Agreement, where WSPL and WS-Agreement provide some support for negotiation. XACML is used to specify access control policies and can be used to support the negotiation of privacy policies [20]. For the specification of general security policies, XACML has a limitation in that its rules cannot be used to define application-dependent concepts such as types of

encryption algorithms. As its policy combinators are implemented subjectively by programmers, the concepts of policy and mechanism become intertwined, which could lead to ambiguities. WSPL is a subset of XACML and can thus support negotiation of mutually acceptable policies by their intersection using combining algorithms. Unfortunately, WSPL has not become a standard.

As the focus of this research is on standard technologies supported by current runtime platforms, this research does not consider more sophisticated languages for negotiation such as WSPL, WS-Agreement, or semantic web policy languages. For B2B web services interactions, it is important to consider WS-Policy and its related specifications, as it is a W3C recommendation since September 2007. Advantages of the WS-Policy framework are that it is flexible and extensible. Policies can be defined inside a WSDL file or defined generically and referenced by any number of WSDL files by making use of reusability mechanisms such as inclusion and grouping of policies. From the perspective of more sophisticated languages, WS-Policy lacks formalisation. Thus, the merging of service consumer and provider policies as means of negotiating an agreed upon security contract is dependent on domain specifications. Such specifications have been defined successfully, but the definition of policy merging and intersection mechanisms needs to be clarified better. WS-Policy is natively supported by common development and runtime platforms.

The negotiation protocol, discussed next, is used to exchange and negotiate a security policy. It should similarly be based on standard technologies to ensure platform interoperability.

### 4.3 Standards-based negotiation protocol

A negotiation protocol is a series of descriptions on how the negotiation is conducted. It is formatted as a set of rules about the interaction manners among the negotiating parties [22]. Generally, automated negotiations can be separated into three main phases [23], namely pre-negotiation, negotiation and post-negotiation.

The pre-negotiation stage begins by starting a new negotiation between two partners. Here, the security policies of partners are

exchanged automatically by the system. The next phase supports the negotiation of the security policy. The two negotiation parties exchange offers and counter-offers for all the security requirements that are being negotiated. Finally, the negotiation process is completed by the creation of the security contract, using the newly negotiated security requirements.

A web services security policy negotiation protocol needs to address rules governing messages, vocabulary, and synchronisation of communication, so that they can be understood by communicating parties. WS-MetaDataExchange [24], [25] defines request-response interactions to exchange policies and other metadata. It has potential to support policy negotiation exchanges using the WS-Policy framework. It is a vendor-independent mechanism for locating and retrieving metadata of a service. For this research, the security policy negotiation system ensures platform interoperability by exchanging standardised security policy documents with a partner until an agreement is reached or the negotiation is aborted. Minimal extensions to this protocol may be required to indicate the status of a security policy document in the negotiation process.

Next, the negotiation strategy is discussed.

### 4.4 Semi-automated negotiation strategy

The traditional form of web services security policy negotiation is performed out-of-band via face-to-face meetings or with e-mail [26]. The disadvantages are that the process is static and time consuming. In a fully automated negotiation process, agents set up, carry-out and finalise the negotiation without any human involvement [20]. As all interactions are machine-based, semantic web technology plays an important role. This approach saves time, and makes the negotiation process very dynamic. For security policy negotiation, this can be risky, as agents are not intuitive and cannot take rational decisions as humans do. The semi-automated approach has the benefits of both the previous strategies in that it is dynamic, uses humans to control some of the decision making, saves time and can be implemented using current web services standards. In a semi-automated negotiation process, agents handle all communications and some decision making. Where conflicts arise, it is resolved by humans who are supported by an intelligent negotiation support system.

For web services security policy negotiation, a semi-automated approach would best match the risk involved and the constraints of the standards-based implementation. Practically, a negotiation process needs to be encapsulated in a negotiation system that incorporates human involvement, which is discussed next.

### 4.5 Negotiation support system

Negotiation Support Systems (NSS) [27] [28] [29] [30] involving humans emerged in the 1980s [31], but were rarely used in practise. NSS normally assist negotiators to weigh up situations, generate and evaluate options, and implement decisions. [32]. There are two main types of NSS. Process-oriented NSS focus on improving the negotiation process, while outcome-oriented NSS help to improve the outcome of the negotiation. For this research, the outcome-oriented NSS is considered, as it supports a negotiation protocol and negotiation help system.

An outcome-oriented NSS utilises the semi-automated negotiation strategy for communication, supported by a negotiation protocol. For the help function, the NSS provides support by structuring and analysing the problem, eliciting preferences and using them to construct a utility function, determining feasible and efficient alternatives, visualising different aspects of the problem and the process [33].

A web services security policy negotiation system needs to include a NSS, possibly consisting of multiple interacting subsystems [34] such as a management subsystem for the negotiation process, a decision support subsystem to advise negotiators, a trust manager to analyse relationships between the negotiators and their respective environments, a conflict resolution subsystem and a contract definition subsystem.

For any negotiation system to be successful it needs to be guided by a formal approach to decision-making, discussed next.

### 4.6 Collaborative decision-making

Two main types of negotiations are distributive negotiations and integrative negotiations [18], [35]. Distributive negotiations are also known as zero-sum or competitive negotiations where negotiators try to

win the negotiation. For B2B web services security policy negotiation, the security contract is thus defined by the security requirements of the negotiator that won.

Integrative negotiations are also known as collaborative negotiations. Collaborative negotiations create a contract between parties in a win-win fashion by finding options that will satisfy both parties [36]. For web services, such negotiations are more likely to succeed, because both parties compromise on certain security policy requirements in order to create a mutually agreeable security contract. The manner in which the negotiation deviates from desired security requirements is dictated by the semi-automated negotiation strategy, characterised by human participation and an intelligent NSS. The inflexible manner in which current web services policy specification languages support policy negotiation is offset by this approach.

Generally, negotiation in e-commerce scenarios focuses on functional parameters of a service such as price, where decision-making usually refers to the process of selecting a particular action in a given situation. Decision models mainly focus on game theoretic models or AI-based models [37]. Such decision-making models do not always incorporate the relationship and influence between different negotiation issues. For example, the choice of an authentication mechanism may be influenced by the level of trust in a partner, as well as the encryption that is supported. Lowering the quality of an authentication mechanism may negatively influence the assurance of non-repudiation. To be able to make decisions that take into account all negotiations issues and their influences, an in intelligent NSS, supported by fuzzy techniques needs to be defined.

For web services security policy negotiation, collaborative decision-making, supported by an intelligent NSS is thus required, in order to negotiate a security contract.

### 4.7 Consideration of environmental influences

The environment in which negotiation is performed has a significant influence on it. Previous research on security policy negotiation has not addressed this issue. There are a number of environmental factors to

consider such as the trust relationship between services, and the dynamically changing context in which negotiation performed.

The level of trust in another service is essential for making rational decisions over the choice of security requirements in an open environment where interacting services often have no previous relationship. Trust relationships evolve by gathering a variety of properties and attributes of consumers, services and other parties, ranging from strong cryptographically verifiable evidence to soft evidence such as a reputation measures and unsigned declarations. An intelligent NSS needs to be supported by a trust manager to assist it with its decision-making processes. Compromises in security policy requirements may be more flexible for highly trusted partners than for strangers.

Consumers and providers should be aware of constraints stemming from a dynamically changing context, which could impact both consumer and provider security policy requirements. For example, if the service platform is under threat, as determined by firewalls and anti-virus programs, it needs a greater the level of protection for the duration of the threat. This will in turn affect decisions that are made during the negotiation. To ensure secure B2B web services interactions, a rapid reconfiguration of security requirements via a context-aware security policy negotiation is needed.

In the next section, requirements discussed here are used to create a framework for the B2B web services security policy negotiation system.

## 5 FRAMEWORK FOR WEB SERVICES SECURITY POLICY NEGOTIATION

In order to negotiate web services security policies, a Policy Negotiation Support Point (PNSP), shown in figure 1, is introduced. The PNSP automatically manages the negotiation, but if a conflict arises, a negotiator makes a decision. The PNSP is knowledgeable about the security requirements of services that can be accessed in its environment, and the standards used by the service and its consumers. Mechanisms exist at providers and consumers to support the publication and exchange of policies. Protocols ensure that messages are sent correctly, so that they are understood by communicating parties. Policies are managed by a policy

manager to guarantee their validity and conformance to standards. The PNSP sources information from a trust manager and platform monitors. The trust manager provides the PNSP with information such as the trust level of the other party and its reputation. Platform monitors are applications such as firewalls, Intrusion Detection Systems (IDS) and anti-virus applications that observe the current environment.

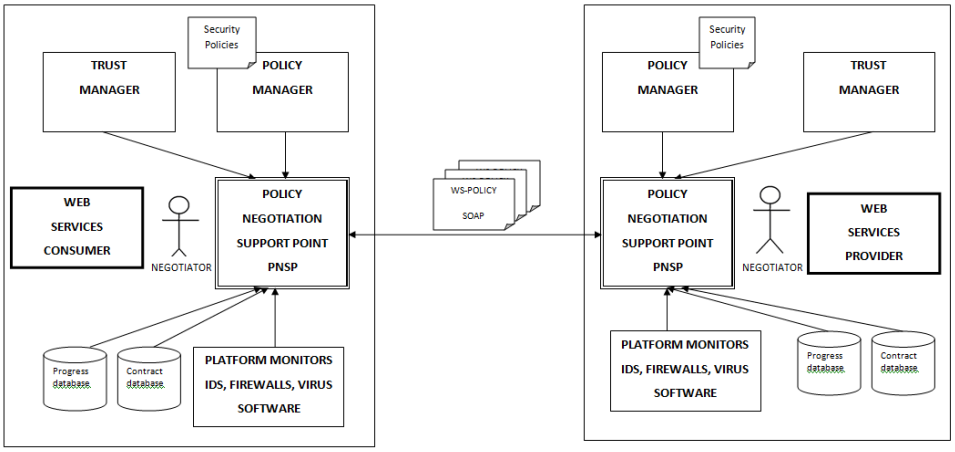


*Figure 1: Web services security policy negotiation system*

A contract database stores security contacts defined during the negotiation process. It contains all previous negotiated contracts, which can be used by the PNSP to make decisions and construct assistance for negotiators. The progress database stores all information about the current negotiation. This allows asynchronous distributed negotiations [13] to take place. As negotiations may take place across different time zones and countries negotiators can negotiate when it suits them.

Both consumers and providers are supported by PNSPs, as shown here. It is also possible that such a component only exists at one party. The first phase of negotiation is started by a consumer and is automatically executed. The consumer sends a request to the provider for the security policy document, defined by the WS-Policy framework. WS-MetaDataExchange request-response interactions are used for this

purpose. The provider's PNSP intercepts the request and returns either the security policy, or a URI (Universal Resource Identifiers) that identifies its location. The consumer evaluates the security policy against its own, to determine whether it would be able to proceed. If there is a direct match of security preferences, a security contract can be agreed to and the service can be accessed without human intervention. If there is not a match, the next phase of the negotiation starts.

In this phase, the PNSP of the consumer attempts to formulate a new offer. It sources information from the trust manager and environmental monitors to determine the level to which a compromise can be made. Different policy choices are generated by the PNSP if possible, and are presented to a negotiator with explanations on how they were determined. The negotiator chooses an option and the PNSP constructs a new offer to be sent to the provider.

The provider receives the offer and if a match can be made, a security contract is created. Otherwise, the negotiation process continues back and forth until both sides agree and the negotiation is successful or one side terminates the negotiation. If the negotiation is unsuccessful, the consumer can search for another service.

## 6 CONCLUSION

Having the ability to negotiate security contracts, web services providers will have the capability to attract more consumers while keeping the security of the web service at an acceptable level. This paper has presented a novel framework for a system that allows the consumers of web services to negotiate a new security contract with the provider of a web service by incorporating environmental considerations into its decision-making.

A list of requirements for a web services security policy negotiation system was defined and analysed. To be able to use the system with current web services technologies, a standards based implementation was preferred, which affected the choice of policy language and communication protocol. The semi-automated approach to negotiation was selected to offset the inflexibility of current policy languages.

Future work aims to investigate the chosen decision model by identifying negotiation objects found in policy documents, and the

environment. Their influence on each other needs to be determined to be evaluated by making use of fuzzy techniques.

## 7 REFERENCES


1. Korba L and Yee G (2008), Security Personalization for Internet and Web Services, International Journal of Web Services Research, IGI Publishing, Volume 5, Issue 1, January-March, pp 1-23.

2. Aref W, Ghafoor A, Joshi J and Spafford E (2001), Security models for Web-based applications, Communications of the ACM, ACM New York, New York, USA, Volume 44, Number 2, pp 38-44.

3. Haas H and Orchard D (2002), Web Services Architecture Usage Scenarios, World Wide Web Consortium (W3C) Working Draft, 30 July 2002, http://www.w3.org/TR/2002/WD-ws-arch-scenarios-20020730 Accessed: 9 June 2009.

4. Booth D, Champion M, Ferris C, Haas H, McCabe F, Newcomer E and Orchard D (2004), Web Services Architecture, World Wide Web Consortium (W3C) Working Group Note 11 February 2004, http://www.w3.org/TR/ws-arch Accessed: 9 June 2009.

5. Duran M and Hasan J (2006), Expert Service-Oriented Architecture in C# 2005, Second Edition, Apress, USA, p 15.

6. Boubez T, Hirsch F, Hondo M, Orchard D, Vedamuthu A, Yalcinalp U, and Yendluri P (2007), Web Services Policy 1.5 - Framework, http://www.w3.org/TR/ws-policy Accessed: 8 June 2009.

7. Hallam-Baker P, Kaler C, Monzillo R and Nadalin A (2004), Web Services Security: SOAP Message Security 1.0 (WS-Security 2004), OASIS Standard 200401, March 2004, http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf Accessed: 9 June 2009.

8. Anderson A (2004), An Introduction to the Web Services Policy Language (WSPL), IEEE 5th International Workshop on Policies for Distributed Systems and Networks, New York, USA, 7-9 June, http://research.sun.com/projects/xacml/Policy2004.pdf Accessed: 8 June 2009.

9. Andrieux A, Czajkowski K, Dan A, Keahey K, Ludwig H, Nakata T, Pruyne J, Rofrano J, Tuecke S, Xu M (2007), Web Services Agreement Specification(WS-Agreement), Grid Resource Allocation Agreement Protocol (GRAAP) Working Group, http://forge.gridforum.org/sf/go/doc14574?nav=1 Accessed: 8 June 2009.

10. Aitken S, Bradshaw J, Dalton J, Jeffers R, Johnson M, Tate A and Uszok A (2004), KAoS Policy Management for Semantic Web Services, IEEE Intelligent Systems, Volume 19, Number 4, July/August 2004, pp 32-41, http://www.aiai.ed.ac.uk/project/ix/documents/2004/2004-ieee-is-uszok-kaos.pdf Accessed: 9 June 2009.

11. Bradshaw J, Jeffers R, Olson L, Tonti G and Uszok A (2004), Integration of KAoS Policy Services with Semantic Web Services, 3rd International Semantic Web Conference, Hiroshima, Japan, 7-11 November, http://iswc2004.semanticweb.org/demos/08/paper.pdf Accessed: 9 June 2009.

12. Debenham J and Elaine L (2008), Automating Contract Negotiation, Fifth International Conference on Information Technology: New Generations, Nevada, USA, 7-8 April, pp 143-148.

13. Kersten G and Noronha S (1997), Supporting International Negotiation with a WWW-Based System, International Institute for Applied Systems Analysis (IIASA), IIASA Interim Report IR-97-049, http://www.iiasa.ac.at/Admin/PUB/Documents/IR-97-049.pdf Accessed: 9 June 2009.

14. Bosse T and Jonker C (2005), Human vs. Computer Behaviour in Multi-Issue Negotiation, Rational, Robust, and Secure Negotiation Mechanisms in Multi-Agent Systems, 25 July 2005, pp 11-24.

15. Cheng W, Lian-chen L, Lung N, Phil M and Wan-cheng N (2007), A Semi-automated Negotiation Process to improve the Usability for Online Marketplaces, 7th IEEE International Conference on Computer and Information Technology, Fukushima, Japan, 16-19 October, pp 253-258.

16. Jang I, Shi W and Yoo H (2008), Policy Negotiation System Architecture for Privacy Protection, 4th International Conference on Networked Computing and Advanced Information Management - Volume 02, Gyeongju, Korea, 2-4 September, pp 592-597.

17. Thompson L (1998), The Mind and Heart of the Negotiator, 1st Edition, Prentice-Hall Inc.

18. Hung P, Jeng J and Li H (2004), WS-Negotiation: An Overview of Research Issues, 37th Hawaii International Conference on System Sciences - Track 1 - Volume 1, Hilton Waikoloa Village, Hawaii, 5-8 January, http://www.uu.edu/personal/hli/index_files/publications/WS-Negotiation.pdf Accessed: 9 June 2009.

19. Brandic I, Buyya R, Mattess M and Venugopal S (2008), Towards a Meta-Negotiation Architecture for SLA-Aware Grid Services, International Workshop on Service-Oriented Engineering and Optimization, Bangalore, India, 17 December, http://www.hpl.hp.com/india/senopt08/papers/senopt08106.pdf Accessed: 9 June 2009.

20. Cheng V, Chiu D and Hung P (2007), Enabling Web Service Policy Negotiation with Privacy preserved using XACML, 40th Annual Hawaii International Conference on System Sciences, Hilton Waikoloa Village, Hawaii, 3-6 January, p 33.

21. Langendörfer P, Maaser M, Ortmann S (2006), NEPP: Negotiation Enhancements for Privacy Policies, W3C Workshop on Languages for Privacy Policy Negotiation and Semantics-Driven Enforcement, Ispra, Italy, 17-18 October.

22. Benyoucef M, Kersten G and Wu S (2006), INSS – A New Approach in Designing Web-based Negotiation Support Systems, Proceedings of the Montreal Conference on e-Technologies, Montreal (Quebec), Canada, May 16-18, http://interneg.concordia.ca/interneg/research/papers/2006/09.pdf Accessed: 9 June 2009.

23. Kersten G and Noronha S (1999), WWW-based Negotiation Support: Design, Implementation, and Use, Decision Support Systems, Elsevier, Volume 25, Number 2, pp 135-154.

24. Samaranayake S (2007), Understanding WS Metadata Exchange - Part I, http://wso2.org/node/2794/print Accessed: 1 May 2009.

25. Ballinger K, Box D, Curbera F, Davanum S, Ferguson D, Graham S, Liu K, (2006), Web Services Metadata Exchange (WS-

MetadataExchange) Version 1.1, August 2006, http://specs.xmlsoap.org/ws/2004/09/mex/WS-MetadataExchange.pdf Accessed: 9 June 2009.

26. Lock R (2006), Automated negotiation for service contracts, 30th Annual international Computer Software and Applications Conference, Chicago, USA, 17-21 September, p 2.

27. Li Y and Shang W (2005), Information Exchange and Conflict Analysis in E-Business Negotiation Support System, 2nd IEEE International Conference on Services Systems and Services Management - Volume 1, Chongqing University, China, 13-15 June, pp 774-779.

28. Archer M, Rose J and Yuan Y (1998), A Web-Based Negotiation Support System, Electronic Markets, Volume 8, Number 3, Routledge, pp 13-17.

29. Dong S, Feng Y and Wang L (2007), The crucial problem of the NSS in the Ecommerce, 2007 International Conference on Intelligent Pervasive Computing, Ramada Plaza Jeju, Jeju Island, Korea, 11-13 October, pp 441-445.

30. Eustice K, Ramakrishna V and Reiher P (2007), Negotiating Agreements Using Policies in Ubiquitous Computing Scenarios, IEEE International Conference on Service-Oriented Computing and Applications, Vienna, Austria, 17-20 September, pp 180-190.

31. Jarke M, Jelassi M and Shakun M (1985), Mediator: Towards a Negotiation Support System, New York University - Department of Information, Operations, and Management Sciences, NYU Working Paper No. IS-85-36, May 1985.

32. Bui TX and Shakun MF (2004), Negotiation Support Systems minitrack, 37th Hawaii International Conference on System Sciences - Volume 1, Hilton Waikoloa Village, Hawaii, 5-8 January, http://csdl2.computer.org/comp/proceedings/hicss/2003/1874/01/187410026.pdf Accessed: 9 June 2009.

33. Kersten G and Lo G (2001), Negotiation support systems and software agents in e-business negotiations, First International Conference on Electronic Business, Hong Kong, China, 19-21 December.

34. Power D (2007), What is a negotiation support system?, Available: http://dssresources.com/faq/index.php?action=artikel&id=137 Accessed: 15 April 2009.

35. Negotiation Types, http://www.negotiations.com/articles/negotiation-types Accessed: 24 April 2009.

36. Nierenburg G (1968), Gerard Nierenberg: The Art of Negotiating, Barns & Noble Books.

37. Faratin P, Jennings N, Lomuscio A, Parsons S, Sierra C and Wooldridge M (2001), Automated negotiation: prospects, methods and challenges, Group Decision and Negotiation, Springer, Volume 10, Number 2, pp 199-215, http://www.csc.liv.ac.uk/~mjw/pubs/gdn2001.pdf Accessed: 9 June 2009.