Business Roles and Negotiation Models for Web Service Based Provision

Thanassis Tiropanis
Intelligent Web Technologies Group
Athens Information Technology Centre, Athens, Greece
ttir@ait.gr

Abstract

The emergence of XML as the lingua franca for communication among applications over the Web, the recent advances in service oriented computing and in web service architectures and the applicability of these technologies in the area of eCommerce necessitates the conceptualisation of business roles and negotiation models for web service based provision. According to the web service paradigm it is envisaged that services will be provided to customers based on dynamic web service composition. This places additional requirements to SLA negotiation in comparison to the traditional service provision paradigm where negotiation for service was performed with a single service provider system. This paper addresses the research challenges with regard to SLA negotiation for web service based provision and outlines business roles and a negotiation model for establishing SLAs with multiple web service providers in order to offer combined web service functionality to match user needs.

Keywords: Web Services, Business Model, SLA, Negotiation, Web Service Orchestration

1. Introduction

The emergence of XML [1] and related protocols allowed software programs to communicate making use of the existing Web infrastructure. Other technologies that already supported distributed computing over Internet protocols have now evolved to support distributed computing over Web protocols and XML. In this way a new paradigm appears in which distributed computing for web services is supported over Web protocols.

Web Services (WS) are distributed, loosely coupled, autonomous software modules that offer specified functionality over Web protocols. The functionality of Web Services can be combined to offer composite services to human users or software applications (Web Service Orchestration). The Web Services community has investigated requirements for WS architectures [2] and provided a number of relevant standards proposing how Web Services can be described, registered, searched for, accessed or composed: WSDL [3], WSFL [4], UDDI [5], etc.

Web Services also support a more general trend towards Service Oriented Computing promising to transform enterprise software systems to an orchestration of loosely coupled reusable service components. Providing a better connectivity among business partners, web services support not only eCommerce in its narrow scope (which by many is perceived as the electronic transactions among business partners) but also eCommerce in a wide scope (which many call eBusiness) addressing all aspects of business operation and support. At the same time, business models for eBusiness/eCommerce have been investigated and categorised [6].

Given these developments the Web can serve as an infrastructure that provides for efficient, user-friendly services by combining available web services to match user needs. This process requires a certain degree of intelligence and the joint work of the Web and the AI communities on building a Semantic Web [7] seems very promising. Proposals on how web services can be intelligently orchestrated to provide composite web services that match user needs have advocated the use of AI planning [8], while the use of semantic web technologies for automating WS discovery and transaction has been supported [9]. However, this alone is not enough. The adoption of a business model for providing and using web services and for supporting charging for WS use can be the driving force for deploying and maintaining high-quality web services.

2. Example of Web Service Based Provision

The following example is to point out the differences between traditional online service provision versus web service based provision. Suppose a customer would like to arrange flight, accommodation and tickets to visit a theme park in California. If these facilities were provided by a single service provider, the customer would visit the provider's web site, review prices, terms and conditions, enter their travel, accommodation and ticket requirements, provide their credit card details and receive confirmation and their tickets by post, fax or electronically. This is based on the assumption that there is a service provider that offers all these facilities in a single service. If this were not the case, such an arrangement would require that the customer interacted with several different services wasting time while being uncertain about the final outcome. This is a traditional service provision case where the provided service (if available) is quite complex, is potentially offering little flexibility to the user and in most cased it is not personalised for individual customer needs.

Considering web service based provision, there will be different autonomous web services each covering an aspect of the service the customer is looking for: web services for purchasing air tickets, web services for booking hotel rooms at the customer's destination, web services for performing secure credit card transactions and web services for purchasing tickets for theme parks. Using AI planning the customer (user) would just express to a system their requirements and the price they are prepared to pay and the system would return a composition of available web services that match the user's requirements. In a web service paradigm where functionality is provided in terms of reusable, loosely coupled components it is much more likely that an orchestration of available web services will match the customer's needs without demanding much personal involvement by the customer. The system however would have to negotiate with each WS provider in this case to check that the total price and the overall terms and conditions match the customer's requirements. Considering the scale of available web services the performance of the negotiation process is critical and it has to be sufficiently addressed before the flexibility of web service based provision demonstrates its benefit.

This paper addresses challenges for web service based provision in terms of negotiation performance and continues to discuss business models and a negotiation protocol to tackle these challenges.

3. Challenges in Web Service Based Provision

In the traditional service provision paradigm where a service is negotiated between a client and a single service provider system there are several requirements in terms of Service Level Agreement (SLA) negotiation. These requirements concern the online performance of the negotiation process, the expressiveness of the negotiated SLAs, interfacing the SLA negotiation process with the rest of the business processes of the service provider: fulfilment, assurance and billing. For this reason, SLA negotiation could involve many iterative negotiation rounds until the service level was agreed by both customer and service provider. However, the provision of a service based on the orchestration of web services provided by more than one web service providers introduces additional requirements.

Using AI planning techniques, a service to support the requirements of a customer in terms of functionality can be mapped onto a number of alternative orchestrations of existing web services (composite web services) provided by different business entities. Before a composite service is available to the customer, negotiation with the provider of *every single* WS participating in the alternative WS orchestrations must be performed. This places strong performance requirements in the Web Service SLA (Web-SLA) negotiation process.

For this reason, alternative Web-SLA negotiation modes should be available that enable (a) *rapid Web-SLA negotiation* when negotiation performance is critical but also (b) *thorough Web-SLA negotiation* in cases where fine-tuning a Web-SLA is required. Available WS provider systems should be able to support both negotiation modes efficiently.

To support reliable and uninterrupted service, should a WS participating in a WS orchestration become unavailable an alternative WS should promptly become available in its place. This requires mechanisms that either perform rapid Web-SLA negotiation with the alternative WS, or activate Web-SLAs that have already been thoroughly negotiated and pre-agreed with the alternative WS provider.

Other challenges concern the ability to identify the WS responsible for failures in cases of composite web services and to collate the Web-SLAs for each WS participating in a composite WS a single SLA document that is meaningful to customers and that clearly distinguishes among the responsibilities of each WS provider.

4. Business Roles for Web-SLA Negotiation

From the viewpoint of negotiation, business roles that are common in different kinds of business models can be identified. Considering a business model classification for eCommerce [6], negotiation for WS use can be highly relevant to the following business models: *Merchant Model*, *Affiliate Model*, *Subscription Model* and *Community Model*.

Provision based on combined web services often demands the formation of a value chain in which different web service providers participate. Very often the customer of a web service is other than the consumer of that service. Therefore, a business model should distinguish among the following roles:

- *Web Service Provider*: A business entity that has the responsibility of making one or more web services available and is able to charge for their use.
- *Web Service Customer*: A business entity that is a customer of a Web Service Provider and can be charged for web service use.
- *Web Service Consumer*: The *human* or *software* user(s) related to a web service customer, which consume the web service facilities offered by a web service provider. The consumers of a web service offered by a provider to a customer are the *consumer domain*. One entity can often implement both the web service customer and the web service consumer roles (e.g. an individual that is a web service user/customer).

Due to the characteristics of web services, being autonomous and reusable software modules that can be orchestrated to offer combined functionality, the provision of services based on a single web service should be considered the exception rather than the rule. The distinction between web service customers and web service consumers is essential to support composite web service provision. This distinction should also be reflected in each Web-SLA between the web service customer and a provider.

5. Web Service Level Agreement (Web-SLA)

A Web-SLA generally describes the commercial relationship between *Web Service Provider* and *Web Service Customer* and the parameters of the web service provision to *Web Service Consumers* associated with the Web Service Customer. Ideally, a Web-SLA will contain all the technical details that will be needed to provide or use web services and, at the same time, it will be easily integrated into a single legally binding document for the provision of a composite web service by a number of WS providers to the customer.

The content of a Web-SLA presents designers with many challenges. Firstly, in order to serve as both a legally binding document and a complete specification of the parameters associated with web service provision a Web-SLA should be expressive and unambiguous. Unambiguity about the obligations and rights of each party in a Web-SLA demands codification in terms of policies to describe rights and penalties when obligations are not observed. A Web-SLA should be able to accommodate both human and software web service consumers and it should be able to efficiently identify consumer domains since the number of consumers related to a web service customer can be large. Supporting secure access for consumer domains to web services presents additional challenges for Web-SLA specification. Further, the ease of integration of Web-SLAs with the business processes of web service providers should be pursued. Finally, it is imperative that each negotiated SLA instance is uniquely identified at each stage of the negotiation process. Currently there is a proposal for a web service level agreement specification language by IBM [10].

6. SLA Negotiation Model

Automated SLA negotiation can take place between a *Web Service Provider Agent* (acting on behalf of a web service provider) and a prospective *Web Service Customer Agent* (acting on behalf of a web service customer). A web service provider should advertise the provided web services and contact details for initiating negotiation. Figure 1 illustrates the SLA Negotiation use case in a UML use case diagram.

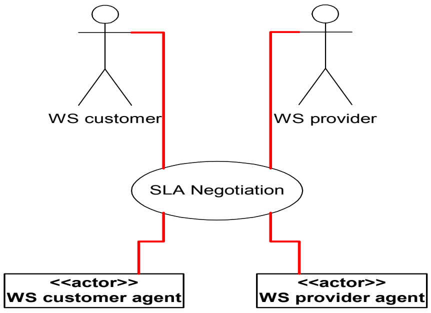

Figure 1. SLA negotiation use case

The design of a Web-SLA negotiation model and a negotiation protocol raises a number of concerns. Electronic Commerce introduces a variety of new business models in which negotiation for web service provision can demand less or more iterations of negotiation rounds before a Web-SLA can be established. For example, in the *Catalogue Merchant* business model [6] a single exchange may be sufficient to establish a Web-SLA for the provision of a service. On the other hand, in the Subscription Model [6] or when auctions are used it can take many negotiation iterations before a Web-SLA is finalised.

The variety of the new business models for eCommerce requires a generic negotiation model while the anticipated large scale of available web services that can participate in negotiation for composite web service provision demands high performance and necessitates design decisions for a Web-SLA negotiation protocol (e.g. stateful vs. stateless, synchronous vs. asynchronous, etc). Using AI planning to provide composite web services a value chain will be formed and negotiated. However, additional to this, back-up web services must be identified should any of the originally selected web services fail. For this reason, a Web-SLA negotiation protocol should provide for either rapid negotiation or for the establishment of pre-agreed Web-SLAs (with backup WS providers) that can be activated by a specified deadline. Finally, secure negotiation is of paramount importance for the design of a Web-SLA negotiation protocol.

Based on this rationale it is proposed that two different kinds of negotiation exchanges are considered:

(a) *Rapid SLA negotiation* where a minimum number of exchanges may take place between a WS provider agent and a WS customer agent. Rapid SLA negotiation can be implemented by an asynchronous stateful negotiation protocol or a synchronous stateless negotiation protocol.

(b) *Thorough SLA negotiation* where SLAs are established following a series of exchanges to form custom agreements between WS providers and WS customers, which can be activated by a specified deadline. An asynchronous stateful Web-SLA negotiation protocol is recommended for this negotiation mode.

When an asynchronous protocol is used, access to a reliable time service will have to be facilitated for the negotiating WS provider and customer agents so that timeouts for message exchanges can be employed; after the defined timeouts the information kept by the stateful agents can be deleted. A model for a negotiation protocol that enables both of the above kinds of negotiation follows.

6.1. Before Negotiation

The recommended negotiation model allows the following possibilities before a SLA negotiation is initiated:

(a) The web service provider advertises a *partially filled out SLA template* that details all the information related to the provision of the web service except for the details of the customer and the consumers (i.e. it specifies provided services, pricing, customer rights & obligations and conditions for use).

(b) The web service provider advertises a SLA *template* that leaves open the customer details, service level and/or pricing information (i.e. it is up to the enquirer to ask for a quote).

The first case provides for *rapid SLA negotiation*, while the second one is best suited for *thorough negotiation*.

6.2. Negotiation Phase

The following sections describe a negotiation round between a WS customer agent and a WS provider agent.

6.2.1. Customer Submitting Proposal

Based on the advertised information, a WS customer agent will have to submit to a WS provider agent a SLA proposal which will be identified uniquely using a SLA_Proposal_ID. The SLA proposal will specify the time by which a response will be expected and if the exchange is to be:

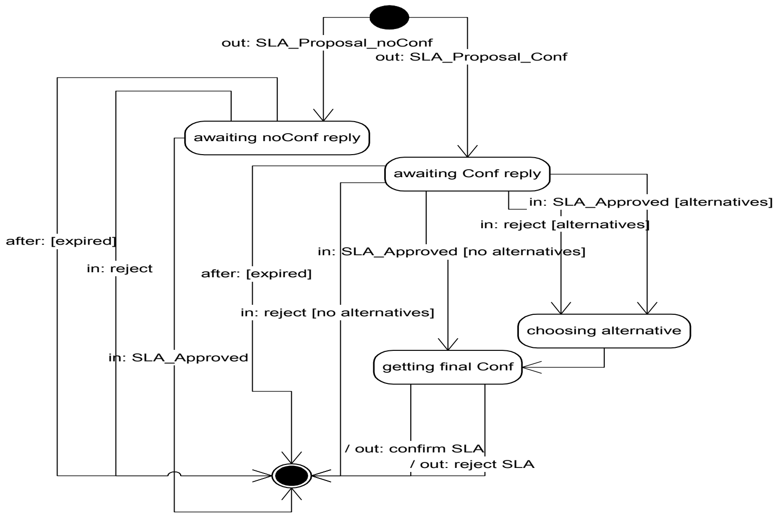

Figure 2. WS customer agent state chart

(a) *A confirmed exchange (for thorough negotiation)*, where the provider's side should expect confirmation from the customer before finalising a SLA (*SLA_Proposal_Conf*). In this case, customer details may be provided in the end with the final confirmation from the customer's side. The SLA templates should always specify what information is obligatory for a proposal.

(b) *A non-confirmed exchange (for rapid negotiation),* where the provider's side, if the proposal is accepted, should finalise the SLA without further confirmation from the customer (*SLA_Proposal_noConf*). In this case, all customer details should be filled in the SLA proposal.

6.2.2. Provider Responding to Proposal

Upon receipt of the proposal the WS provider can respond in the following ways:

1. Reject the proposal as syntactically incorrect.
2. Reject the proposal based on its content. For example, if the proposal is incomplete, if the customer/consumers are out of service area, if the requested levels cannot be accommodated or other reasons.
3. When thorough negotiation is requested (confirmed exchange), reject the proposal based on its content but counter-propose a number of alternative SLAs (SLA_Alternative_Approved) with parameters that can be accommodated by the WS provider. Each alternative will be identified with a SLA_Alternative_Approved_ID and will reference the SLA_Proposal_ID provided by the customer. Each alternative will also specify a deadline by which it will be valid.
4. Accept the proposal as it stands sending its approved form to the customer (SLA_Approved). It includes a unique SLA_Approved_ID and references the related SLA_Proposal_ID.
   (a) For thorough negotiation (confirmed exchanges), the time by which customer confirmation is due will be specified. Further, the WS provider agent may optionally suggest alternative SLAs (as in item 3 above).
   (b) For rapid negotiation (non-confirmed exchanges) SLA_Approved will be finalised by the provider and the negotiation round ends.
5. Not reply to the SLA proposal of the customer letting the proposal expire.

6.2.3. Concluding Negotiation.

Upon receiving one of the above responses, the WS customer agent can respond in the following ways:

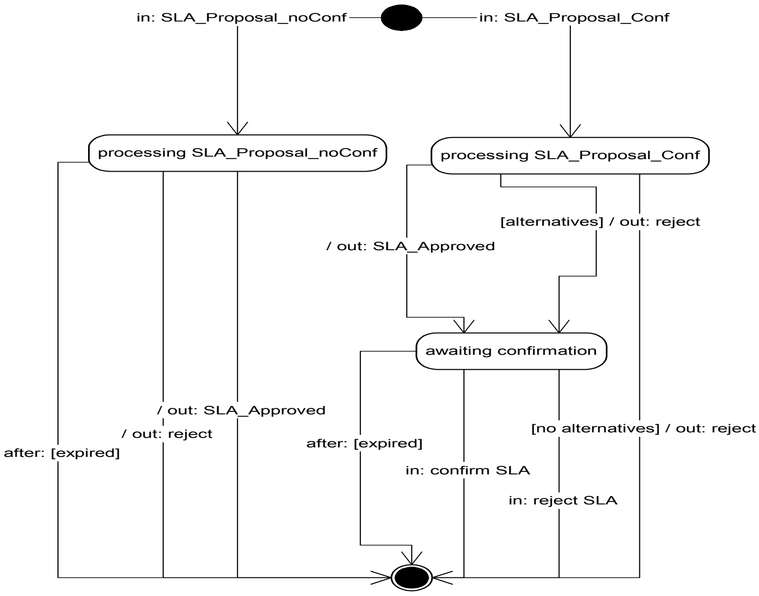

Figure 3. WS provider agent state chart

1. If the proposal is rejected as syntactically incorrect the negotiation round ends.
2. If the proposal is rejected based on its content and no alternative SLA is proposed negotiation ends.
3. If the proposal is rejected but alternative SLAs are proposed by the provider's side, the WS customer agent can accept one of the proposed alternative SLAs (providing the corresponding alternative agreement ID), or reject them altogether (quoting the SLA_Proposal_ID), or let them expire.
4. If the proposal is accepted and SLA_Approved is provided by the provider's side, then:
   (a) For rapid negotiation (non-confirmed exchange) the agreement is considered final.

(b) For thorough negotiation (confirmed exchange), if no alternative SLAs are attached, the WS customer agent will accept the agreement, potentially subject to final confirmation by the customer, or reject it. The SLA_Approved_ID will have to be referenced in either case.

(c) For thorough negotiation (confirmed exchange), if alternative SLAs are proposed, the WS customer agent can pick one of the proposed agreements (providing the corresponding alternative agreement ID), or reject them altogether (quoting the SLA_Proposal_ID), or let them expire.

In (b) and (c), customer details should be provided with the confirmation if they were not submitted with the proposal.

5. If the SLA_Proposal expires without response from the provider's side the negotiation round ends.

Upon confirmation of acceptance of a SLA (SLA_Approved or a SLA_Alternative_Approved) from the customer by the specified deadline, the WS provider agent will close the negotiation round and the SLA will be activated. If no SLA is agreed, thorough negotiation allows for additional negotiation rounds where the customer can submit new SLA proposals based on the outcome of the previous rounds. The UML state charts of Figure 2 and Figure 3 illustrate the operation of the WS customer and provider agents respectively.

7. Conclusions and Further Work

The previous sections presented the promise of web service based provision and supported the necessity for the definition of business roles and a model for Web-SLA negotiation to enable the dynamic and efficient establishment of business relationships among actors involved in web service based provision. The anticipated scale of web services necessitates the use of AI planning techniques for web service orchestration and thus, a Web-SLA negotiation protocol that provides for both rapid and thorough negotiation is proposed.

The next step of this research is to further investigate how Web-SLA negotiation can be best integrated into the planning process based on this protocol and to minimise the burden that negotiation adds to the planning process. The definition of Web-SLAs that can efficiently express rights and obligations in terms of policies and which can be integrated with the business processes of web service providers is critical, while supporting secure access for web service consumer domains to web services remains a challenge.

References

[1] Eds: T. Bray, J. Paoli, C.M. Sperberg-McQueen, E. Maller, "Extensible Markup Language (XML) 1.0 (Second Edition)", W3C Recommendation, 6 Oct 2001

[2] D. Austin, A. Barbir, C. Ferris and S. Garg, "Web Services Architecture Requirements – W3C Working Draft", WWW Consortium, 14 Nov 2002

[3] Eds: R. Chinnici, M. Gudgin, J.J. Moreau, S. Weerawarana, "Web Services Description Language (WSDL) Version 1.2, W3C Working Draft, 2 Mar 2003

[4] F. Leymann, "Web Services Flow Language (WSFL 1.0)", IBM Software Group, May 2001

[5] T. Bellwood, et al, "UDDI Version 3.0", UDDI Spec Technical Committee Specification, 19 Jul 2002

[6] E. Turban, D. King, J. Lee, M. Warkentin, H. Michael, C. Chung, "Electronic Commerce: A Managerial Perspective", Prentice Hall, 2002

[7] T. Berners-Lee, J. Hendler and O. Lassila, "The Semantic Web", Scientific American, May 2001

[8] S.A. McIlraith, T. Cao Son, and Honglei Zeng, "Semantic Web Services", IEEE Intelligent Systems, March/April 2001

[9] M. Paolucci, K. Sycara, T. Kawamura, "Delivering Semantic Web Services", Proceedings of the World Wide Web 2003 Conference (WWW2003), Budapest, Hungary, 20-24 May 2003

[10] H. Ludwig, A. Keller, A. Dan, R. P. King and R. Franck, "Web Service level Agreement (WSLA) Language Specification version 1.0", IBM, 28 Jan 2003