2008

A Framework for Automatic SLA Creation

Halina Kaminski
University of Western Ontario, hkaminsk@csd.uwo.ca

Mark Perry
University of Western Ontario, mperry@uwo.ca

A FRAMEWORK FOR AUTONOMIC SLA CREATION

Halina Kaminski and Mark Perry
Department of Computer Science
The University of Western Ontario, London, Ontario, CANADA

ABSTRACT

Negotiation is fundamental to business. Increased automation of business to business or business to customer interaction is demanding efficient but flexible systems that can manage the negotiation process with minimal direct human intervention. Industries that provide online services rely on Service Level Agreements as the basis for their contractual relationship. Here we look at a means for generating these with a negotiating tool (SLA Negotiation Manager) that complies with e-negotiation rules and creates the agreements from existing business objectives.

KEYWORDS

Service Level Agreements, SLA management, Automated-agreements.

1. INTRODUCTION

The changes in today's business processes are very rapid. Service based enterprises found their way to form the base of current Information Technology market. Service Level Agreements (SLAs) are commonly used to ensure business success and customer's satisfaction. This paper describes a negotiation protocol for an Automated Negotiation Manager. In most information technology service offerings, the critical components and the level of delivery must be identified and agreed upon. Such steps must include development of Service Level Agreements (SLAs) between a service provider and its customers. Every SLA is prepared and 'signed' by all parties involved in an agreement to assure customers that they will get the service that they pay for, and define the limits of the obligations placed on the providers. Very often there are serious financial consequences for the provider for not meeting the SLA specifications, therefore companies that provide services have a natural interest in understanding their capabilities to deliver the highest quality performance that they can. At the same time Service Level Agreements should match business needs of both sides as close as possible.

2. BACKGROUND

In [1], *negotiation* is described as an iterative communication and decision making process between two or more parties who:

- Cannot achieve their objectives through unilateral actions;
- Exchange information comprising offers, counter-offers and arguments;
- Deal with interdependent tasks; and
- Search for a consensus which is a compromise decision.

There are two possible outcomes of a negotiation: a compromise or a disagreement. Traditional negotiations for years have been performed face-to-face. Recently, a large number of such negotiations have moved on-line forming a well structured e-negotiation system that involves Negotiation Support Systems and Negotiation System Agents [1]. SLA Negotiation Manager is intended to be a Decision and Negotiation Support Tool in SLA contract creating process.

Most of today's research in the area of providing computing services concentrates on how to manage the SLA compliance as well as tracking performance for planning purposes [2],[3]. This work proposes

an automated system that creates the SLAs and concentrates on developing a negotiating tool (SLA Negotiation Manager) that will automate the development of SLAs which are based on existing SLOs, existing SLAs and existing business policies. Earlier work in SLA management has focused on a bottom up approach, looking to capture manage SLA data [4]. Currently, most contract negotiations are done offline and in a presence of executive personnel. Through the use of automation, the cost loads of managing personnel can be reduced [5]. The development of an SLA negotiation system will provide a great asset to service provision enterprises.

This work approaches the problem with the steps and activities from the service provider side of negotiations, since that is where the SLAs usually originate, and the most useful structure for the implementation of an automated system. From the provider's point of view the SLA should achieve its business specifications and at the same time it should maximize the customer's satisfaction. SLA development should be considered as a vital step in the business process. The most recommended strategy to prepare good SLAs is to set Service Level Objectives (SLOs).

Here we present the automation of the SLA creation process from a set of Service Level Objectives (SLOs), employing software agents and adopting a utility function by incorporating it into the decision process. A utility function helps to determine what will be the negotiation outcome. By adopting this system, the service provider can form SLAs and satisfy the need for fast and flexible agreements. Earlier work in SLA management has focused to capture managed SLA data [2]. However, this study concentrates on automatic SLA creation that integrates an effective negotiation process, removing the need for the service provider to engage highly qualified personnel at the time of SLA adoption by the customer. One area in which companies are seeing increased cost is support personnel for their system offerings. Where a company's business is primarily (software) service provision, such costs are critical to contain. Such environment make it very desirable to automate the monitoring, selection, and decision making processes, leaving the service provider more resources to focus on the provision of better services and reducing support and management costs [3].

Many of the business decisions are based on resource prioritization that in turn helps to increase the company's profit. In this paper a resource means any service that is quantifiable, such as application, server, CPU usage, disk space, license etc [6]. The automation can be achieved by building a software system that embodies high level decisions and which possesses the properties of autonomies (software agent exercises exclusive control over its actions), social ability (agent has ability to reason), reactivity (sensing and perceiving change in an environment and responding) and pro-activeness (ability to improve knowledge and performance through learning) [7]. Software agents can provide this type of functionality, and an SLA real-time negotiation system that utilizes these features will prove to be a great asset to service provision enterprises.

3. SERVICE LEVEL AGREEMENT

A Service Level Agreement (SLA) is intended to be a formally written agreement between service providers and their customers. The contents of an SLA may vary for different services but there are basic clauses including QoS requirements and penalties if QoS requirements are not satisfied. It is not always the case that the QoS requirements provided by a service for a specific consumer are the same for each use of that service by the consumer.[8] In between the use of the service by the consumer the resources needed to support the QoS requirements of the first invocation of the service may not be available for the second invocation. One possible reason is an increase in the number of consumers and the demand on the system. This suggests that SLAs should be generated on demand. It has been proposed that each service provider should generate service offers with terms that the provider can support. The consumer can then choose the offer that best satisfies his needs.

SLAs are typically meant to be negotiated and agreed upon prior to the running of B2B (business to business) applications. In fact, they are usually the end product of the negotiation process, and represent a form of a contract for monitoring and compliance purposes. Once the QoS parameters are negotiated and agreed upon between parties, these parameters can be manually entered in a tool which generates a static SLA document. The existing practices include also using a word processing tool to document an SLA. The SLA is then deployed on the system, and typically a third party is asked to monitor its compliance.

As per SLA specification document, a fixed (non-negotiable) SLA can be published on a custom generated Company's Registry for all clients. Such registry can be generated using a data base utility which could store SLAs in a machine readable format (such as an XML). In order to maximize the functionality of SLA, the proposal is to develop the SLOs containing performance expressions using WSLA language. The prototype application will demonstrate this concept by allowing a client to discover a number of different SLOs that are defined for a specific service.

4. THE AUTOMATED NEGOTIATOR

The integration of Web Services and aims embodied in the autonomic computing paradigm require the ability to manage business objectives for the service provision and service level agreements through translation into service level objectives and their management as global resources. In addressing the research problem, this paper aims to provide the means for abstraction and selection of objectives from agreements and aims. The best strategy to prepare effective SLAs is to set level objectives that support business needs. Usually, the services provided by an IT company vary both in diversity and complexity, thus planning and documenting services is critical. [1] says that automated e-negotiations can be successful if they follow a well structured protocol. Such negotiations employ software agents that make decisions and control the entire process, including the specification of offers and concessions, and the final decision about agreement or disagreement.

An automated SLA creation system will eliminate much inefficiency caused by lack of resource specific knowledge. By using templates and SLO libraries, the SLA Negotiation Manager will ease the contract creation. The system will follow a closed negotiation protocol meaning that new rules can not be added throughout the negotiation process. The negotiation medium will provide a user-friendly interface for the client to see and choose requested services. Contract creation time will be reduced by using the templates and pre-approved clauses. By using the system, the service provider will ensure consistency and compliance with company's standards. The SLA Negotiation Manager embodies the business knowledge, goals, and policies of the implementing party. Such knowledge enables the system to choose and combine the set of SLOs that should be specified in the SLA in order to ensure compliance with the business goals. The system will provide the compliance monitoring according to customers choices. The essential work in creating SLOs takes place in the business/marketing department, as it is here that the responsibility lies in gathering and analyzing business objectives from the needs of the outside market environment. It is very important that business decisions are based on resource prioritization. The existence of a number of service levels and performance metrics for each resource results in multiple SLOs for every service. Generally, each objective can include parameters such as average response time, system availability, transaction rate, validation time period and the cost of proposed performance. The validation time period has to be specified during the negotiation phase i.e. when the customer and the service provider agree to the specific service terms.

The SLA resulting from the SLOs should be targeted to give users the best possible performance for each service level defined. In addition to the customer and provider identification information, the SLA should specify the service that will be available to the customer. There are many different types of SLAs, ranging from the very basic to precise, focused SLAs that vary from customer to customer in the same enterprise [9].

5. SOFTWARE AGENTS, NEGOTIATIONS, UTILITY FUNCTION

In order to use business policies as an input they have to be specified in a machine readable language. There are some readily available standards to specify business policies (such as: WS-Policy, Policy RuleML, Business Rules Markup Language – BRML, Simple Rule Markup Language – SRML, etc.). We assume that we will be able to evaluate and choose the most suitable one for our engine.

The most challenging part of our project is to choose a set of algorithms that given SLOs, SLAs and Business Policies as an input it would assess an existing data and as a result it would produce the best

service offer to the customer in such a way that it maximizes provider's profits and at the same time maintains customer's satisfaction.

Hopefully, with some modifications, some of the probabilistic auction negotiation algorithms can be used. This needs to be researched more, but an idea is that a Negotiation Manager will behave as an Intelligent Agent who will have knowledge about the system and it will use that knowledge in order to come up with the best offer. Here the project could benefit from the use of the Ontology Management System [10] to define semantics of the resources available for services offered. Another potential candidate to underlay the negotiation process is basic Game Theory [11],[12]. Although it seems to satisfy the main requirement of decision-making based on multiple input it must be thoroughly examined before it can be said it can be used.

Automated contract creation will enable service providers and their clients to make use of technology to create SLAs within pre-planned and pre-approved parameters. We aim at the automation of SLA development and creation, which is synonymous with electronic contracts for computing services, from SLOs. In addition to giving flexibility the contracting system will optimize the provider's profits at the same time as maximizing the customer's satisfaction. The proposed system will be built in such a way that it will negotiate on behalf of service provider. The overall negotiation process will be modeled as exchanging proposals and counter-proposals between the provider and the customer. At the beginning provider needs to come up with an offer pack that is based on service, price, delivery, quality etc. For that a set of SLOs can be used.

The provider takes all factors into account and calculates the expected pay-off value function associated with possible offers, and selects the offer that maximizes his payoff. If the customer is satisfied with an offer, he just sends an acceptance message to the provider and an SLA is finalized. If the client does not accept the first offer, then he can either abort the negotiations or he can send a counter - proposal. At this point the provider evaluates an offer and updates its knowledge about the customer.

If the offer is acceptable the Negotiation Manager creates an SLA, otherwise provider sends counter-proposal. Exchange of counter-proposals goes until one of the parties decides to accept an offer or quit.

The management of conflicting preferences in a group of parties is one of the major research areas in multi-agent systems. One of the major concerns is a social choice problem. According to [13] the social choice problem is to find a function that fairly aggregates conflicting preferences. We consider a service provider and service consumer to be the negotiating parties. Very often customer satisfaction becomes secondary to resource utilization and profit maximization. By using a utility function we would like to enable the negotiation medium to take customers' preferences into account. The function will help to make rational decisions under incomplete information. In a negotiation system some subjective assumptions often need to be made because the service provider has limited knowledge about the customer and his preferences.

We would like to use a utility function that will describe decision maker's 'compromise' profile and will incorporate his attitude toward specified decisions. To be able to define a utility function we need to define a utility scale to measure the subjective value of dollars under given conditions. On an interval utility scale, each unit measures equal magnitudes that the service provider might use to calculate the next offer. For the purpose of this project we have decided to set one percent as a single unit. For example, if the customer does not agree with the offer, next offer might introduce 5% off the initially proposed price. Subsequently, every next offer can carry a fraction of this discount. The value and the usage of the utility function depends on business objectives and can be different for each service provider/customer relationship.

Negotiation Manager Model

An Automated Negotiation Manager model is a 7-tuple: {R, K, Z, P, Q, F, M} where:

R is a set of participants. This set contains all parties that can be involved in the contract. The customer, service provider and all supporting parties belong to this set. At least two elements of this set (service provider and customer) must participate in any SLA negotiation process qn $\in$ Q.

K is a set of all possible agreements (SLAs) created using set P. Every existing SLA agreement that is stored in a data base belongs to the set K. It also contains a set of possible agreements (limited by P) that can be created as a result of any successful negotiation process.

Z is a set of business rules (also called business knowledge). A business rule that a service can not cost less than $0.07 per transaction might be an example of zi \in Z. Set Z represents corporate preferences and aligns business strategies of a service provider.

P is a set of company defined SLOs. Every SLA contains at least one SLO for the agreed service.
Q is a set of all sequences s, such that every s =q1,q2,q3 … qn where qi is an action (an offer, a counteroffer, accept or decline).
Each s illustrates a negotiation process and every successful negotiation is a finite sequence s. Here, by successful negotiation we mean any negotiation process that resulted in either accept or decline. Sequence s can also serve as a history log when stored in a repository. The past negotiation procedure can be recreated from such sequence.
F is a utility function. This function is customized according to the negotiating party needs and business preferences. For example it might be widely known that the customer offers 10% less for the service than he is really willing to pay. Function f might be used to calculate next offer: f = current offer * 10%.
M is a set of all possible offers that can be created using set P. Every permutation of elements of P belongs to M. In addition M contains any combination of an offer that has been modified according to one or more business rules from set Z.
There have been many mathematical models developed for negotiations, typically on direct e-commerce negotiations, and often employing game theory algorithms. Although these are not directly applicable to the SLA environment where there are a great deal more factors to consider above the product and price, they are useful for further development of the negotiation system.
To ensure stability of resources offered only one client might be negotiating at any given time. In this system the software agents are **not** called intelligent because they have no ability to learn client's behaviour during the negotiation process (the intelligent systems emulate the human ability to perceive, reason, make decisions and act). The knowledge of the system such: business rules, number of SLAs, resource specific information presented in SLOs, is given at the initialization time.
Having a wide variety of offered services poses another difficulty. To create a good SLA from a pool of many different SLOs one must think thoroughly of the resource dependencies and connections between resources. For example, having two servers that are capable of handling 1000 transaction per second each does not necessary mean that we can safely agree to provide a service of 2000 transactions per second to the customer. Both of these servers might be using a secondary resource that is limited to the lower capacity. The overall performance is hardly ever a simple summation of the resources available.

A. Use case scenario

Online Tales is a Web based book store. It is looking for a web service that will help the owners to perform sales transactions with the online customers. It is required that the service will be available 90% of the time. The initial response time has to be lower than 10 seconds. CompuSale is a company that provides Web services. It specializes in supporting sale transactions of goods, and most of its customers are the online stores. Online Tales staff examined UDDI directory where they found a service offering from CompuSale. Online Tales decided that the offering can be modified if CompuSale could lower the price.

6. PROTOCOL

Step 1: Initiation

Initial offer:	
Service:	*sale transaction*
Response time:	*less than 10 seconds*
# of transactions per second:	*100*
Availability:	*90%*
Price:	*$50- per day*

Master Agent waits for the customer to connect. When the connection is made, the Master Agent determines what is the requested service (resource) and who is the client involved in a negotiation process. Master Agent delegates this knowledge to the software agents responsible for Business (Business Agent), SLOs (Resource Agent) and SLAs (Contract Agent).

Step 2: Investigation

Individual Agents investigate their own domains in order to limit the utility function to high utility contract regions. Agents come back with a range of values which are needed and to be used in negotiation.

Business Agent returns with the information about the customer and the resource. There are several customer types in a system. Each customer belongs to one of the following groups: new, bronze, silver, gold, platinum, or restricted. The groups are created by the business (marketing) people. For example a customer who belongs to the silver type is the one who uses our services for at least one year, or restricted customer is the one whom we do not want to make any deals with. In addition to the customer information, a business agent brings the information about the requested resource. For the resource, the agent determines a set of attributes that can be negotiated, atomic unit value, and a unit utility value for each attribute. For example if the attribute is a response time, for each unit of increased response time (say unit =1 second) the utility value might be 5% deducted from the final price presented in an offer. Every resource has its minimum price assigned to it. This is the value that under no circumstances, should the Master agent offer the price lower than this value for that particular resource. It is in the best interest of a provider, that such values are determined and set by business personnel who probes the market and comes up with the best possible strategy. Business agent merely gathers information, and returns it to the Master Agent.

Business Agent returns with the following values:
CustomerType = New
Attribute 1: Availibility , attribute_unit =1; Attribute_Unit value = 3%
Attribute 2: # of Transactions per second, attribute_unit =10; Attribute_Unit value = 5%
Attribute 3: Response time, attribute_unit =1; Attribute_Unit value = 7%
Minimum price = 30 per day

Resource Agent is responsible for the information hidden within a pool of SLOs. This agent takes a resource id as an input (it does not have to know the customer name) and finds existing SLOs for the resource. This agent gets a set of negotiable attributes and values associated with each attribute. For example the resource can have a response time negotiable with following values associated with it: 1, 2, 5, and 8. Service availability might be another attribute for this resource with values: 95, 90, 85, and 80. Resource agent will gather only the values lower than the value presented in an initial offer. It is assumed, that the customer is willing to sacrifice quality for the lower price. It is also anticipated that the best possible quality has been offered via UDDI directory to begin with. Resource agent also returns SLO id for each vale of the attribute. This is done to make the Contract Agent's task easier when creating SLA. Final SLA will consist of the SLOs chosen by the customer to fit his/her needs. Having SLOs identifications readily available will prove to be handy when SLA needs to be created.

Resource Agent returns with the following values:
Attribute 1: Availability, values: 85(id23), 80(id25), 75(id33), 70(id14)
Attribute 2: # of Transactions per second: 90(id77), 80(id12), 70(id42), 60(id51), 50(id 6), 40(id34), 30(id68), 20(id3), 10(id26)
Attribute 3: Response time: 11(id17), 12(id89), 13(id57), 14(id19), 15(id39)

Step 3: Presentation

Agents present their findings to the Master Agent. A set of negotiable attributes is created and communicated to the customer. Customer is asked to identify the attribute(s) that are going to be non-changeable. For example a resource has three attributes: response time, resource availability and number of transactions per second. Customer is asked to chose which attributes are important and shall remain unchanged. In this example customer might chose the response time as the most important and not to be changed, therefore resource availability and number of transactions per second become negotiable attributes. Customer can chose up to (n – 1) attributes to be static, since the initial offer already presents the price for n attributes.

CompuSale Service Offer #1: sale transactions

Response time:	*less than 10 seconds*
# of transactions per second:	*100*
Availability:	*90%*
Price:	*$50.-*

Online Tales can choose one or two from the following attributes to be not changeable:

- *Response Time*
- *Number of transactions per second*
- *Availability*

Online Tales chooses Response Time and Availability to be static

Step 4: Evaluation

Master Agent evaluates its utility by summation (aggregation) of values of constraints. Based on information provided by Business Agent and Resource Agent, the Master Agent finds the common utility space. It calculates every attribute's value and modifies the price by the utility value of that attribute. For example if the attribute offered was a response time of 1 second, and the next available value for that attribute is 3 (with the atomic unit = 1 second, and the utility value = 5%) then the final offer will be adjusted to the following: [initial offer - (2 units * 5%)]. In other words there will be 10% deducted from the final price presented in a next offer. The evaluation takes place for each negotiable attribute.

Step 5: Decision

Master Agent finds the best possible combination. Master Agent generates an offer. Offer is presented to the customer.

CompuSale Service Offer #2: sale transactions

Response time: less than 10 seconds
# of transactions per second: 90
Availability: 90%
Price: $47.50
Calculation:
# of transactions unit = 10, Attribute_unit value = 5%
Initial offer: 50

*Next Offer =50 – (50 * ((100 – 90)/10 * 5%) =50 – 2.5 = 47.5*

CompuSale Service Offer #3: sale transactions

Response time: less than 10 seconds
# of transactions per second: 80
Availability: 90%
Price: $45.-
Calculation:
# of transactions unit = 10, Attribute_unit value = 5%
Initial offer: 50

Next Offer = 50 – (50 * ((100 – 80)/10 * 5%)) = 50 - 5 = 45.-

Step 4 and 5 are repeated until Master Agent can no longer find any values that lie within min price boundaries or customer terminates negotiations, or the time expires. After each iteration: initial offer = last offer

Step 6: Agreement

Both parties agree on the terms, or terminate negotiations.
When Online Tales accepts an offer the time for the contract is determined. The offer price is multiplied by number of days of the Agreement.

Step 7: Finalizing

Contract Agent is initiated in this step. It has the least work assigned to it. Contract Agent does not come into the picture until the end of the negotiation, and is activated only when an agreement is reached upon which the customer is willing to sign in for the service. Then SLA contact needs to be created. Contract Agent has the knowledge of SLA language constrains, and creates an agreement according to the XML based WSLA standards. The final SLA is presented for the final acceptance, and when agreed upon, it is stored in SLA database.

7. CONCLUSION

It is our goal to provide an automated way to create Service Level Agreements. At the same time we would like to improve ways to enhance the negotiation process and the quality of agreements. It is likely that there will be billions of WS invoked daily, through millions of machines or agents, working on behalf of humans. WS availability, reliability, and scalability continue to be a center of the SOA research. Our work offers a unique approach to the SLA creation mechanism. The packaging of SLOs, SLAs, and business policies files would allow the customer to evaluate and choose the best offer available. Extensions to this research would include examining social order functions to match the objectives of Negotiation Manager, and of course further implementation to validate the model in a more rigorous form. Most of the functions available today have no notion of business policies in them. It is possible that taking an existing function and modifying it to take business policies into an account will provide a function that could be used in a service system on application servers.

AKNOWLEDGMENT

The authors thank the Natural Science Research Council of Canada for their support.

REFERENCES

[1] Bichler, Martin, Kersten Gregory, Strecker Stefan, “Towards a Structured Design of Electronic Negotiations”, Group Decision and Negotiation, 2003, Vol. 12, No. 4, (pp 311-335)

[2] Faratin P., Jennings N.R., Lomuscio A., Parsons S, Sierra C, Wodridge M. “Automated negotiation: prospects methods and challenges.” International Journal of Group Decission and Negotiation, 10(2):199-215, 2001.

[3] Beam Carrie, Segev Arie “Electronic Catalogs and Negotiations”. CITM paper 96-WP-1016 http://haas.berkley.edu/~citm/wp-1016-Summary.htm

[4] Buco M.JU., Chang R.N., Luan L.Z., Ward C., Wolf JL., Yu P.S., “Utility computing SLA management based upon business objectives” IBM Systems Journal Vol 43 No.1 2004 p.159.

[5] Suh, Bob, “Avoiding an Austerity Trap” Outlook Journal, February 2004 Retrieved from: http://www.accenture.com/Global/Research_and_Insights/By_Subject/High_Performance_Business/AvoidingtheAusterityTrap.htm on Dec 12, 2005

[6] Mark Perry, Michael Bauer, “Policies, Rules and their engines: What do they mean for SLA?” Proceedings of Knowledge-Based Intelligent Information & Engineering Systems Conference KES 2004 Wellington, New Zealand September 2004

[7] Faramarz Farhoodi, Fingar, Peter, “Competing for the Future with Intelligent Agents”. Distributed Object Computing, “DOC” Magazine, November 1997

[8]] Dan, A., Ludwig, H., Pacifici, G., “WS differentiation with SLAs,” Retrieved May 11, 2006, http://www-128.ibm.com/developerworks/webservices/ library/ws-slafram/

[9] Asit Dan, et. al., “WS on Demand: WSLA-driven automated management,” IBM Systems Journal, Jan. 2004

[10] Juhnyoung Lee , Richard Goodwin , Rama Akkiraju , Yiming Ye , Prashant Doshi. Ontology Management System.30 October 20003 http://www.alphaworks.ibm.com/tech/snobase

[11] Dafaalla Hussam, “Agent-Oriented EAuctions: Market Services for e-marketplaces”, Master Thesis, Faculty of Graduate Studies, University of Western Ontario, 2003, p 12.

[12] Binmore, Ken, Vulcan Nir, “Applying Game Theory to Automated Negotiation”. DIMACS Workshop on Economics, Game Theory and Internet, April, 1997

[13] Brant, Felix, “Social Choice and Preference Protection”, EC’03, June 9–12, 2003, San Diego, California, USA. ACM 158113679X/03/0006.