A Generic Negotiation and Re-Negotiation Framework for Consumer-Provider Contracting of Web Services

Werner Mach University of Vienna Faculty of Computer Science Vienna, Austria werner.mach@univie.ac.at

ABSTRACT

Electronic contracting is key issue for establishing liquid markets dealing with electronic goods. This paper presents a framework for automatic negotiation between Web services. The major goal of the framework is comprising all necessary components for negotiation and re-negotiation. The capabilities and the components are described both for Web service consumer and provider. The re-negotiation framework is based on independent control loops and associated knowledge bases for consumer and provider implementing specific business strategies. An economic cost model considering variable cost and a business rule repository are main parts of the knowledge bases. Using the concept of autonomic managers gives the framework the ability to act as self governing system. Furthermore the framework can handle auctioning by using predefined workflows as so called auctioning plug-ins.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous; H.3.5 [Online Information Services]

General Terms

Design, Economics, Theory

Keywords

Auction and Negotiation, Service Level Agreements, Business Strategy, Agents, Web Services

1. INTRODUCTION

The importance of electronic contracts is crucial for future e-Business models due to the increasing importance of Web services and the cloud as a reliable commodity providing business value. Negotiation is the prerequisite for a successful contract between two ore more partners. These contracts are usually based on Service Level Agreements

IIWAS '12 Bali, Indonesia Copyright 2012 ACM ...\$0.00. Erich Schikuta University of Vienna Faculty of Computer Science Vienna, Austria erich.schikuta@univie.ac.at

(SLAs). As described in [10], service providers can make use of SLA technology to advertise and offer their services capabilities while consumers are able to formalize their service level objectives through SLAs. A Service Level Agreement is a formal negotiated agreement and an explicit contract between a provider and a consumer of a service. A SLA is defined through a negotiation process between consumer and provider [20].

These contracts are described in protocols like WS-Agreement (Web Service Agreement) [1]. The WS-Agreement specification is a standardization effort conducted by the Open Grid Forum (OGF) in order to facilitate creation and monitoring of SLAs. It is a simple request-response protocol for agreement creation and monitoring.

WS-Agreement has become a standard for providing electronic contracts. The reason for the success of WS-Agreement is the flexibility and ability for using it in automated service integration in cloud based systems. WS-Agreement defines the involved partners, the context of the agreement and describes the necessary guarantees of the defined service levels.

In this paper we distinguish between negotiation and renegotiation. Negotiation is defined as the process defining SLAs, re-negotiation is the process altering existing SLAs, e.g. in case of a SLA violation. Negotiation is an proactive process while re-negotiation is an reactive process.

The main goal of both involved parties is to create and operate SLAs with a minimum of human interaction, but also to negotiate and agree upon legally binding electronic contracts. Balancing these objectives is a non trivial task. Both of the parties, consumer and provider, have their own business rules implementing a specific business strategy to make their decisions during negotiation and if necessary correction phase. Business rules are stored in the knowledge base of the framework and during negotiation all decisions are based on it.

This situation leads to the idea to design, develop, and implement an automatic negotiation framework and build upon an economic knowledge base using a novel cost model considering variable cost. This cost model allows to define own business strategies depending on easy understanding parameters. To the best of our knowledge no ICT cost model (specifically for cloud computing) exists that makes a distinction between variable and fixed cost [3, 12, 22, 2, 4, 6]. Cost models that consider variable cost are better suited to to derive more accurate business strategies.

Traditional business models are largely based on fixed cost operating models. This operating models are driven of large capital investments to leverage economies of scale to pro-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

duce incremental profit in case of increasing volume. The result is spreading the operating cost to larger and larger sold units. But this models only work if the products are stable. The prediction of sales of the products are stable enough to give the possibility to allow companies to allocate labor and capital to support the demand. Variations of the demand can only be compensated for low frequencies. Typical product life cycles in the past were measured in years, therefore this kind of operating model can be used. Today product life cycles are shortened from years to month and even hours. The predictability of such markets is much more difficult and more complex. Also rapidly evolving consumer preferences in global markets need more flexible cost models to give quick answers to changing demands.

The contribution of this paper is the definition of a novel negotiation and re-negotiation framework which allows for automatic commerce of Web services based on economic principles. This enables market-based service trading (following a bazaar style) and extends the classical supermarket approach typical for service negotiation today. Therefore our approach extends the WS-Agreement standard by feasible workflows to supports auctioning for negotiation and re-negotiation. A specific highlight of our framework is the mapping of business strategies defined by economic goals of the respective organization into an ICT enabled framework. It facilitates autonomic agents acting as organizational representatives stipulating service level agreements without human interaction. This allows for business transactions transparently to the environment but adhering to business objectives of the originating organization (i.e. company, industry, community, etc.).

The paper is structured as follows: Section 2 describes the challenges of the service market today and motivates our research. In section 3 we introduce self governing systems and present our novel contracting framework describing in detail all comprising components and processes. Section 4 exemplifies our approach by a practical scenario and shows the feasibility of our framework. The paper is closed by a conclusion and remarks on future work.

2. CHALLENGES IN THE SERVICE MAR-KET

An important characteristic of Service Marketplaces is the liquidity of the traded good. A sufficient number of market participants is necessary for the proper function of the market. Resource providers and consumers are only willing to join, if on the one hand the provider can be fairly certain that its resources will be sold, and on the other hand the consumer will be able to find what it needs quickly. An open market approach enables providers to create a wide range of resource types and allows consumers to specify their needs precisely. The SLAs differ often slightly only. Simulations performed by the authors [15] show that large resource variability of both sides results into a large number of resource offers. Therefore the probability of matching decreases rapidly. But to ensure sufficient liquidity in the market the probability should be high. The matching probability can be used as a measure to determine how attractive a market would be to providers and consumers.

In recent years, a large number of commercial cloud providers have entered the utility computing market, offering a number of different types of services. There are resource providers who only provide computing resources like Amazon or Tsunamic Technologies and SaaS providers who sell their own resources together with their own software services (e.g. Google Apps, Salesforce.com). Typical providers only sell a single type of resources (with the exception of Amazon). This limited number of different resource types enables a market creation, since all demand is channeled towards very few resource types.

To fully exploit the potential of open markets, a large number of providers and consumers is necessary. But the large number of potential traders might inflate the variety of resources which leads to the problem that the supply and the demand are spread across a wide range of resources. To give traders few restrictions, an approach is needed which allows traders to define their resources (or requirements) freely while facilitating SLA matching. Current adaptive SLA matching mechanisms are based on semantic ontologies like Web Ontology Language OWL [5] and OWL-S (former DAML-S) and other semantic technologies. However, none of these approaches addresses the issues of the open market and deals with (semi-)automatic definition of SLA mappings enabling negotiations between inconsistent SLA templates. Specifically focussing cloud computing no existing framework for re-negotiation in clouds includes a cost model based on a knowledge base. None of these considers business rules to give answers to typical business strategy questions. Conclusively a model which considers both, energy- and costefficiency doesn't exist.

3. A FRAMEWORK FOR CONTRACTING OF WEB SERVICES

In current SLA research the negotiation between provider and consumer is not or only insufficiently considered. In most cases *one-phase negotiations* are being used to keep the effort for the negotiation small. One-phase negotiation means service providers offer their services in the form of agreement templates. A template may contain a number of alternative service descriptions with different service quality. The consumer can choose one of these templates which fulfills its requirements best. After choosing the offered template consumer and provider create an agreement. This approach is like buying in a supermarket: the provider offers a set of products and the consumer chooses one or more of it.

In [17] Negotiation and Re-negotiation are defined as:

Negotiation is a process between an agreement initiator and an agreement responder to reach an acceptable agreement offer from an initial agreement template. Agreement offer negotiation is a non-binding, bi-lateral process that comprises exchange of information in order to find a consensus for acceptable agreement offers.

Renegotiation is a process between an agreement initiator and an agreement responder to reach an acceptable agreement offer in order to alter an existing agreement.

Open Grid Forum has defined an extension to the WS-Agreement Specification [1], the WS-Agreement Negotiation version 1.0 [17] which allows multi-round negotiation necessary in many scenarios. However, the WS-Agreement Negotiation protocol does not explicit support auctioning and

biding. Nevertheless, the protocol can be used for communication. Auctions and biding are complex negotiation processes [18]. Peter Wurman et.al. [23] developed an internetbased platform for price-based negotiation - the Michigan Internet AuctionBot. It was designed to serve as an auction server for humans as well as software agents. But nevertheless auction protocols are one-to-many negotiations and are out of scope of the WS-Agreement Negotiation model. Auction protocols require alternative negotiation approaches.

3.1 Self Governing Infrastructures

Existing frameworks for self-governing ICT (information and communication technology) Infrastructures use a knowledge base for their decisions during operation. Self-governing principles augment the autonomic systems. In autonomic systems the rules and policies are defined by humans whereas self-governing systems may produce, improve, and evolve the rules without intervention from outside.

Due to dynamics of infrastructure changes (e.g., frequent service failures) the rules for QoS re-negotiation have to evolve reactively (e.g., new negotiation strategy have to be used). This has to happen without human interaction and has to be based on predefined guidelines.



Figure 1: Concept of an Autonomic Manager

The concept of an autonomic manger for self governing systems 1 lists four states: Monitor, Analyze, Plan and Execute (MAPE). All actions taken are based on a knowledge base. Sensors for the managed resource are necessary for monitoring and actuators are used for changing the managed resources.

The knowledge base of our framework consist of a *Business Rules Repository* which includes decision-trees and - tables and a *Economic Cost Model* considerating variable cost and energy efficiency.

3.2 Components of the Framework

In the following we introduce a generic framework for both consumer and provider and describe all necessary components.

As depicted in figure 2 the framework is symmetric; each component exists for both service provider and -consumer. Please note that framework supports also many-to-many associations between consumer and provider. This is necessary for auctioning and biding.

The **Negotiation and Re-Negotiation Engine** is one of the main components of our framework. This engine is responsible for both negotiation and re-negotiation. That means for the whole life-cycle of a Service Level Agreement. The negotiation engine is designed as an autonomic manager. The engine uses the knowledge base for all their decisions during life cycle of each SLA. To extend bilateral contracting an *Auctioning Plug-in* is used to support also auctioning and biding. The plug-in can be easily replaced to support different types of auction models. The sensors are responsible for monitoring the agreed QoS of the running services and applications. Actors can alter already running services.

The Knowledge Base consists of three parts, the Business Rules Repository where the participant (consumer or provider) store their own business rules which implement the specific business strategy of the participant. The repository uses decision trees and tables. The second part of the knowledge base is the economic cost model. In this cost model the cost of each production factor like computational power, disk space and network bandwidth are stored. The cost model distingishes between fixed cost and variable cost. This is necessary to give correct answers for a decision e.g. if a provider have free capacity of a resource and a customer already running a service is paying the fixed cost and the provider want to decide at which price the free resource can be offered. Existing cloud cost models neglect traditional economic principles. Cost models that consider variable cost are better suited for our economy situation today. This distinction is also necessary to implement or derive more accurate business strategies.

The economic cost model as knowledge base in the framework has to support all traditional economic fundamentals and methods described in the standard economic literature [16]. Traditional economics covers operating production factors, the production, sales theory, and investment and finance. This issues are necessary to allow applying all well known traditional economic methods to the cost model. In [14] we developed a comprehensive cost model based on variable and fixed cost for common cloud computing environments applying traditional economic methods. Based on this model we show that business strategies for both cloud providers as well as cloud consumers can be derived. On the one hand this model gives the ability to design new cloud computing environments and also optimize existing clouds; on the other hand this model can also be used to give a clear answer for building internal cloud environments to IT Managers.

The third part of the knowledge base is *history data*. The history data stores past information. That means the experience each participant has made in contracts and running services, e.g. how many successful contracts have been done between consumer and provider, statistical data about the QoS of each provider, etc. This information is very useful for decisions during negotiation and re-negotiation.

The **Service Template Registry** is the a registry where providers can store their offered services. The consumers can



Figure 2: Framework Overview

retrieve this registry for finding the best matching service they need. The Service Template Registry is used during first time establishing agreements. That means standard offers from the provider are stored. During Re-Negotiation the registry is not been used.

The **Consumer- and Provider Agents** are the interfaces between the framework and an application or a human person. The interface is for initiating and controlling services, for changing running services, maintaining the knowledge base and for the auctioneer controlling of and interacting with the auctioning workflows.

The **Auctioneer Agent** is controlling and interacting the auctioning workflow stored in the *Auctioning Plug-in*.

The **Protocols** used in the framework adhere to existing standards: WSDL for service description, WS-Agreement for order and offer description in the service template registry and WS Agreement Negotiation for the contracting process. The WS-Agreement Negotiation model defines the

negotiation as a separate process. During this process service consumer and provider exchange information dynamically with the goal of creating a valid agreement offer that subsequently leads to an agreement. Generally negotiation takes place prior to service execution. Re-negotiation is a reaction of one of the parties on the actual performance of the service execution.

3.3 Framework Pattern

The framework supports two patterns for establishing agreements. The first one, the **n:m Negotiation Pattern**, is negotiation and re-negotiating directly between consumer and provider. Figure 3 depicts negotiation in which only consumers and providers participate. In this scenario $n_1 \ldots n_4$ service consumers and $m_1 \ldots m_3$ service providers are involved. The Service Template Registry is used by all service providers and consumers to advertise SLA-templates respectively to retrieve it. Between consumer and provider is an n:m relationship, that means each consumer can have a relationship to any of the provider.

The second framework pattern (figure 4) is the **Auction**ing Pattern, where one or more *Auctioneers* are involved. The negotiating workflow (i.e. the auctioning type) is stored in the *Auctioning Plug-in* of the Auctioneer, Consumer and Provider Agent.

It is important to note that the framework supports each kind of mixture of the two negotiation types.



Figure 3: n:m Negotiation Pattern



Figure 4: Auctioning Pattern

In the following we use the UML to give one exemplary sequence that can be realized in this scenario.

Figure 5 shows a possible negotiation flow of the first type of negotiating (see figure 3). In this negotiation process four roles are involved. The service template registry acts as a mediator between providers and consumers. It is responsible for finding a matching provider for the consumer. The provider creates offers and the consumer searches for them. Both, consumer and provider, use a proprietary knowledge base for analyzing and creating offers. In figure 5 the communication between the knowledge base KP1 and provider P1 is simplified.

- 1. Provider P1 advertises his offer on service template registry S.
- 2. The consumer C1 looks at this registry S for matching providers. The registry suggests provider P1.
- 3. This provider creates the offers O1 and O2 based on knowledge base KC1 and sends these offers to the consumer.

- 4. As the consumers knowledge base advises to reject O1 the consumer sends a reject message to the provider. For offer O2 the consumer sends a counter offer.
- 5. The provider analyses the counter offer and sends a new offer based on the consumers counter offer.
- 6. The consumer checks this new offer and rejects it which triggers the provider to create and send a new offer based on the customers counter offer.
- 7. As the consumers knowledge bases considers this offer as acceptable the consumer sends an accept message to the provider which replies with an agreement.

Figure 6 depicts the sequence diagram of the second type of negotiating scenario, auctioning (see figure 4). In the auctioning based negotiation process four roles are involved. The auctioneer is responsible for the whole process including defining and controlling the auctioning rules, collecting all offers and bids, executing clear actions, etc. The provider creates offers and the consumer searches for them. Both, consumer and provider use a knowledge base for analyzing and creating offers.

- 1. **Offer Phase**: First of all the provider *P*1 creates an offer using its knowledge base and sends it to the auctioneer *A*.
- 2. Bid Phase: Auctioneer A collects all bids from the consumers C1 and C2. The consumers create their bids based on their knowledge bases.
- 3. **Quote Phase**: The auctioneer A accepts a bid (the quote) applying auction rules (e.g. Vickrey auction). After a quote is fixed, the provider is informed which creates a template for the agreement based on the offer.
- Clearing Phase: The winning consumer C1 receives an accept message from auctioneer A and the agreement template from provider P, consumer C2 receives a reject message.

Figure 7 presents a code snipped of a WSLA document, which describes this situation. It shows a service level objective for hard disc space which is part of the obligation section. In the example provider P1 is the obligated party for the time period defined in the validity block. The expression section contains the content of the obligation. The obligated party must provide HardDiscSpace from 30 to 90, whereas HardDiscSpace is a SLA parameter defined with an appropriate metric in a separate section of the document.

To show the different states of each participant we present a state diagram of both provider and consumer in our framework.

Please note in our framework consumer and provider have the same state diagram. That means they execute the same states and event triggers. In figure 8 all states and trigger events are depicted are described as follows:

• Running

In this state normal or idle operation is performed, e.g. running services after successful contracting.

• Negotiating

After the initial state the negotiating state can be



Figure 5: This picture shows a running example of a possible negotiation flow.



Figure 6: The figure illustrates a sequence diagram for negotiation with auctioning



Figure 7: The figure illustrates a code snippet of a WSLA document



Figure 8: State diagram for consumers and providers

reached. The decisions necessary for negotiating are based on stored business rules and the embedded cost model.

• Initiate services

After successful negotiating the services are initiated or scheduled if start time is different.

• Re-negotiating

The re-negotiating state is only reached if a SLA violation happens or the consumer wants to re-negotiate because another provider offers a more attractive service. In this state this is the chance for both consumer and provider to avoid executing the contracted SLA penalties. As in the negotiation state the decisions necessary for negotiating are based on stored business rules and the embedded cost model.

• Execute SLA penalties

This state is reached if the re-negotiating state failed for successful contracting.

4. A PRACTICAL SCENARIO

In this section we present a practical scenario using our generic framework due to the fact of lacking a working implementation yet. The development of our framework is ongoing and we plan simulations on the running prototype in the near future.

In the following we will focus on the realization issues of the two central components of our envisioned re-/negotiation framework: on the one hand the knowledge base of the business rules repository implementing the company's business model; and on the other hand the re-/negotiation process for agreeing on service level agreements implementing the web service contract.

4.1 Business Model

For the course of simplicity of the scenario we restrict to risk management issues, which are defined by the strategy of an enterprise' business model using a cost model and additional business rule.

Generally, a company coping with risks, has to choose a basic risk management strategy, as *Accept*: Accept the risk without taking countermeasures; *Avoid*: The operation causing the risk will be discontinued; *Migrate*: Take countermeasures for the risky operations; *Transfer*: Transfer the risk to one other organization by e.g. covering an insurance.

Our framework is able to apply all kind of risks in the business rules repository. In our example we use *Migrate* as a risk management strategy type. Figure 9 shows a risk model coping with the cost for risks migration [13]. c_{dis} represents the cost of disruption. A disruption causes two "types of cost": The disruption cost, c_{dis} , are cost which are tangible such as e.g. contractual penalty or loss of income. The additional cost, which are difficult to measure such as damage of reputation or loss of potential customers, are of the second "cost type". These cost aren't considered in disruption cost, c_{dis} .

Countermeasures should prevent and reduce the risk to an acceptable limit by reducing the time of disruption and reducing the disruption cost c_{dis} consequently [11]. However, the cost for these counter measures and the security cost c_{sec} are increasing with rising efficiency. In our example, counter measure costs are represented by security cost c_{sec} . The cost minimum, c_{min} , represents the best balance between the disruption cost, c_{dis} , and the security cost, c_{sec} , where the total cost c_t are minimized.[13]

As mentioned above, the disruption costs c_{dis} do not consider reputation cost. However, companies paying attention to their reputation claim a higher security level, e.g. banks. Therefore the framework business rules applying this model are able to consider a "security markup". The customer can ask for such a security markup representing the loss of reputation in case of disruption. The security markup helps



Figure 9: Risk model

to approximate the disruption cost c_{dis} to the real cost and leads to increased total cost $c_{min_{sm}}$. This approximation is represented by the disruption cost considering a security markup $c_{dis_{sm}}$. Our cost model delivers the disruption cost (penalty cost). One of the business rules in our business rule repository contains the security markup cost.

Table 1: Risk model parameters

acronym	description
c	Cost
c_t	Total costs
sm	Security markup
c_{min}	Cost minimum
$c_{min_{sm}}$	Cost minimum considering security markup
c_{sec}	Cost for security
c_{dis}	Cost for disruption
$c_{dis_{sm}}$	Cost for disruption considering security markup

For the implementation of the business model we use a rule based approach. Specifically we plan to apply Rule Responder¹, which is a rule-based enterprise service middleware for distributed rule inference services and intelligent rule-based Complex Event Processing on the Web. It weaves the outer shell of our business model realization by providing the required infrastructure for the automation of role description of partners as well as steering and redirection of the distributed queries and SLA management. This proven approach (see our work on rule-based SLA management [7, 9]) comes also very handy for the second component of our framework, the re-/negotiation component, where the result of a contracting process is defined by a set of SLAs.

4.2 Contracting Process

For the following contracting process we assume a negotiation process as described by our *Auctioning Pattern* in section 3. Hereby we differentiate 4 phases according to figure 6: Offer Phase, Bid Phase, Quote Phase, and Clear Phase.

4.2.1 Offer Phase

The providers present their offers by defining security measures with respective cost. In our example we assume that the the provider is following "Pareto Optimum" offers in accordance to its business model. A "Pareto Optimal Front" describes all offers for which the improvement regarding one objective automatically leads to a worsening of at least one of the other objectives. In our case it is simply interpreted as: a decrease of costs reduces the security [19]. Thus the provider presents a set of Pareto optimal offers to the auctioneer as starting point for the auction's partner, the consumer agent. For the identification and description of the Quality-of-Service attributes of the provisioned services we developed a classification and ontology framework for distributed and heterogeneous Systems [21].

4.2.2 Bid Phase

In figure 9 c_{min} on the total costs graph represents the break even point, giving the maximum security for minimum cost. Depending on the company's strategy the expenditures of possible bids follow the c_{dis} (minimizing security, mimizing costs) or the c_{sec} (maximizing security, maximizing cost) elasticity trajectory. In this example we assume a bank company with a high security markup, which means that only offers below the elasticity trajectory to the right of the break even point are acceptable. Thus as a first step in the Bid Phase the bank as provider is looking for offers within this designated offer area. If there are more than one offer available the one is chosen fitting best to the business strategy of the consumer, i.e. in the case of a bank maximizing security up to a defined cost limit.

4.2.3 Quote Phase

After acceptance of the bid by the consumer the quote is set by the auctioneer. Now the agreement between provider and consumer is defined using a formalized SLA methodology. Hereby we apply our proven SLA manipulation methodology [8], which allows to define agreements on a formalized basis and to map them to respective WS-Agreement descriptions.

4.2.4 Clearing Phase

In this phase all auction partners are informed about the auction result. In our case validation, penalty and reward agreements have to be defined and put in place. These activities are realized by our rule-based SLA management framework [7, 9].

5. CONCLUSION

Due to the shift of economy from CAPEX to OPEX organizations and businesses in future service-oriented infrastructures need to act in a more agile fashion than ever before. Thus the ICT environment has to adapt automatically to changing needs. Building value chains based on Web services is becoming increasingly important. However, until now, no framework exists which realizes an electronic marketplace where service provider and service consumer can trade their goods easily and automatically.

In this paper we introduce a novel negotiation and renegotiation Framework consisting of a knowledge-based re-

¹http://responder.ruleml.org

/negotiation engine. The knowledge base has a built-in economic service cost model and a business rule repository. Trading decisions require support by a system implementing the business strategy of the participating business partners by delivering all relevant business information such as estimated fixed costs and variable costs. Such a cost model is the necessary requirement for automatic negotiation and re-negotiation of services. Our novel framework allows to implement the central business goals for both service consumer and service providers. Using the concept of autonomic managers gives the framework the ability to act as a self governing system. To support all possible kinds of auctioning we design a changeable auctioning plug-in as a part of the negotiation engine. For establishing SLA's a service template registry is used where consumer retrieve templates and provider store their templates. For communication between the components we use standard protocols like WS-Agreement and WS-Agreement Negotiation. As already laid out in the paper we establish a methodological basis by combining several findings of our latest research, as cloud-enabled economic and energy-aware cost models [14], SLA-management [8], QoS ontologies [21], and rule-based business knowledge and process management [9, 7].

The implementation of the framework is ongoing. In the near future we will conduct an empirical evaluation of our approach by simulation. A main focus of our future research will be an analysis of the behavior of different auctioning models. Specific focus will be laid on defining business strategies in the knowledge base of the autonomic system. Thus, the framework aims for automatic, adaptive, and dynamic negotiation and re-negotiation processes establishing an ICT marketplace for services.

6. **REFERENCES**

- A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu. Web services agreement specification (ws-agreement). In *Open Grid Forum*, 2007.
- [2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009.
- [3] R. Buyya, R. Ranjan, and R. N. Calheiros. Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities. *CoRR*, abs/0907.4878, 2009.
- [4] R. N. Calheiros, R. Ranjan, C. A. F. D. Rose, and R. Buyya. Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services. *CoRR*, abs/0903.2525, 2009.
- [5] F. v. H. D. McGuinness. Owl web ontology language overview. world wide web consortium. http://www.w3.org/TR/owl-features/.
- [6] A. G. Greenberg, J. R. Hamilton, D. A. Maltz, and P. Patel. The cost of a cloud: research problems in data center networks. *Computer Communication Review*, 39(1):68–73, 2009.
- [7] I. U. Haq, R. Alnemr, A. Paschke, E. Schikuta, H. Boley, and C. Meinl. Distributed trust management

for validating sla choreographies. In *SLAs in Grids Workshop as part of Grid 2009 Conference*, October 2009.

- [8] I. U. Haq, A. Huqqani, and E. Schikuta. Hierarchical aggregation of service level agreements. *Data and Knowledge Engineering*, 70(5):435–447, 2011.
- [9] I. U. Haq, E. Schikuta, I. Brandic, A. Paschke, and H. Boley. SLA validation of service value chains. In 9th International Conference on Grid and Cloud Computing (GCC'10), page 308–313, Nanjing, Jiangsu, China, 2010. IEEE Computer Society.
- [10] P. Hasselmeyer, H. Mersch, B. Koller, H. N. Quyen, L. Schubert, and P. Wieder. *Implementing an SLA negotiation framework*, volume 4, pages 154–161. IOS Press, 2007.
- [11] B. D. Jenkins. Security risk analysis and management. *Countermeasures*, 1998.
- [12] D. Kondo, B. Javadi, P. Malecot, F. Cappello, and D. P. Anderson. Cost-benefit analysis of cloud computing versus desktop grids. In *IPDPS*, pages 1–12, 2009.
- [13] R. Lepofsky. Quantifying risk and cost of it security compliance. SecurityActs, 3:11–14, May 2010.
- [14] W. Mach and E. Schikuta. A consumer-provider cloud cost model considering variable cost. In *IEEE Ninth International Conference on Dependable, Autonomic* and Secure Computing, DASC 2011, 12-14 December 2011, Sydney, Australia, pages 628–635, 2011.
- [15] W. Mach and E. Schikuta. A bazaar simulation of web service negotiation. submitted for publication, 2012.
- [16] N. G. Mankiw. *Principles of Microeconomics*. Thomson South Western, 2004.
- [17] OpenGridForum. Ws-agreement negotiation. http://www.ogf.org/documents/GFD.193.pdf, 2012.
- [18] S. Parsons, J. A. Rodriguez-Aguilar, and M. Klein. Auctions and bidding: A guide for computer scientists. *ACM Comput. Surv.*, 43(2):10:1–10:59, Feb. 2011.
- [19] C. J. Petrie, T. A. Webster, and M. R. Cutkosky. Using pareto optimality to coordinate distributed agents. Arti Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM, 9:269–281, 1995.
- [20] A. Pichot, O. Wäldrich, W. Ziegler, and P. Wieder. Dynamic sla negotiation based on ws-agreement. In WEBIST (1), pages 38–45, 2008.
- [21] E. Vinek, P. P. Beran, and E. Schikuta. Classification and composition of qos attributes in distributed, heterogeneous systems. In 11th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid 2011), Newport Beach, CA, USA, May 2011. IEEE Computer Society Press.
- [22] B. Wickremasinghe, R. N. Calheiros, and R. Buyya. Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications. In AINA, pages 446–452, 2010.
- [23] P. R. Wurman, M. P. Wellman, and W. E. Walsh. The michigan internet auctionbot: A configurable auction server for human and software agents. In *In Proc. of* the second International Conference on Autonomous Agents, May 1998.