Negotiation of Service Level Agreements: An Architecture and a Search-Based Approach

Elisabetta Di Nitto[1], Massimiliano Di Penta[2], Alessio Gambi[1], Gianluca Ripa[1], and Maria Luisa Villani[2]

[1] CEFRIEL - Politecnico di Milano
Via Fucini, 2 20133 Milano
[2] RCOST - Research Centre on Software Technology
University of Sannio – Palazzo ex Poste, Via Traiano 82100 Benevento, Italy
dinitto@elet.polimi.it, dipenta@unisannio.it, alessiogambi@gmail.com, ripa@cefriel.it, villani@unisannio.it

Abstract. Software systems built by composing existing services are more and more capturing the interest of researchers and practitioners. The envisaged long term scenario is that services, offered by some competing providers, are chosen by some consumers and used for their own purpose, possibly, in conjunction with other services. In the case the consumer is not anymore satisfied by the performance of some service, he can try to replace it with some other service. This implies the creation of a global market of services and poses new requirements concerning validation of exploited services, security of transactions engaged with services, trustworthiness, creation and negotiation of Service Level Agreements with these services. In this paper we focus on the last aspect and present our approach for negotiation of Service Level Agreements. Our architecture supports the actuation of various negotiation processes and offers a search-based algorithm to assist the negotiating parts in the achievement of an agreement.

Keywords: Quality of Service, Service Level Agreements, Negotiation, Optimization Heuristics.

1 Introduction

Software systems built by composing existing services are more and more capturing the interest of researchers and practitioners. The envisaged long term scenario is that services, offered by some competing providers, are chosen by some consumers and used for their own purpose, possibly, in conjunction with other services. If the consumer is not anymore satisfied by the performance of some service, s/he can try to replace it with some other service. This implies the creation of a global market of services and poses new requirements concerning validation of exploited services, security of transactions engaged with services, trustworthiness, creation and negotiation of Service Level Agreements (SLAs).

This paper focuses on SLA negotiation. While this issue has been deeply studied within the domain of e-commerce, there are not many approaches that

B. Krämer, K.-J. Lin, and P. Narasimhan (Eds.): ICSOC 2007, LNCS 4749, pp. 295–306, 2007.

focus specifically on the domain of services. In such a context, the subject of the negotiation is the definition of so called SLAs, that is, more or less formal contracts that discipline the way services are provided to consumers and, in turn, the obligations to be fulfilled by the consumer in order to obtain the service. Such SLAs can either be negotiated on a per service usage basis, or they can have a longer term validity. This last one is actually the most common situation, but the other should be possible as well, even if to be effective it requires a fast execution of negotiation.

Negotiation can either be performed directly by the interested stakeholders or it can be automatic. In this second case, human beings are replaced by automated negotiators that try to achieve the objective that has been suggested to them. Automated negotiation is particularly important when the consumer of a service is a software systems that has to negotiate on the fly (part) of the SLA with the service. In the following we present our approach for negotiation of SLAs. Our architecture supports the actuation of various negotiation processes (one to one negotiations, auctions, many-to-many negotiations) and offers an efficient search-based algorithm to assist the negotiating parts in the achievement of an agreement.

The paper is structured as follows. Section 2 provides some definitions that will be used through the rest of the paper. Section 3 presents an overview of the architecture of our system. Section 4 focuses on the negotiation search-based approach while Section 5 presents some preliminary simulation results that show the advantages of this approach. Finally, Section 6 provides a comparison with the related literature and Section 7 draws the conclusions.

2 Definitions

According to Jennings *et al.* [6], a negotiation can be defined as: *"the process by which a group of agents come to a mutually acceptable agreement on some matter"*. We argue that a negotiation process requires the following key elements:

1. *The negotiation objectives*, i.e., the set of parameters over which an agreement must be reached. These can include the price of the service usage, its availability, the nature of the operations the service will make available, etc.
2. *The negotiation workflow*, i.e., the set of steps that constitute the negotiation; they depend on the kind of negotiation that is actually executed (bilateral bargaining, auctions, reverse auctions, etc).
3. *The negotiation protocol*, i.e., the set of conditions that indicate the validity of all information concerning the negotiation and provided by the negotiation participants. For instance, if the adopted negotiation process is an English auction, the negotiation protocol will define as acceptable only those offers that improve the values associated to the negotiation objectives.
4. *The agent decision model*, i.e., the decision making apparatus the participants employ to act in line with the negotiation protocol in order to achieve their objectives. For example, this can be based on (i) the acceptable ranges for the negotiation parameters (definition of sub-domains); (ii) functions to

evaluate the offers; (iii) the goal to pursue, e.g., maximize one ore more utility functions; and (iv) a strategy to pursue that goal, that is, the algorithm to decide the moves, in reply to the move by some other participant.

The agents may use both a *cooperative* or *competitive* approach to come to an agreement. This is determined by the kind of the interdependence of the respective interests [5] and has an impact on the process they follow to come to an agreement.

In a Service Oriented Architecture (SOA) context, the negotiation participants are essentially service providers and service consumers, although an overparts mediator could be included to provide conciliation mechanisms or help setting up cooperative strategies. In automated negotiations, (part of) these participants are replaced by software agents that act on behalf of them.

Negotiation objectives, workflow, protocol, and decision models depend on the multiplicity of the participating agents:

- **1-1:** it is the most known approach and requires that a consumer bargains with a provider for the definition of the SLA of the specific service;
- **1-N:** a consumer bargains with a set of providers. These providers can either compete among each other to reach the agreement with the consumer (this approach is applicable when the consumer needs to obtain a binding to a single service) or they can cooperate to share the service provisioning, e.g., for example, split the service availability interval.
- **M-1:** several consumers bargain with a single provider. Consumers in this case can either compete to acquire an SLA with the provider (in an auction style) or they can obtain a single SLA with the provider to share its resources (e.g., the bandwidth).
- **M-N:** combining the previous two multi-party negotiation types, at one side, service providers could cooperate to reach an integrated SLA. On the other side, consumers may fight to get the best Quality of Service (QoS) guarantees for that service.

Differently from many works in literature that support specific negotiation processes, we aim at developing an infrastructure that can be tailored depending on the multiplicity, workflow, protocol, and decision model that fit a specific application domain. Furthermore, we exploit optimization techniques to speed up the search for agreements in (semi)automatic negotiations.

3 Negotiation Architecture

The architecture of our negotiation framework, shown in Figure 1, is composed of a *Marketplace* and various *Negotiation MultiAgent Systems*, each associated with a specific negotiation participant, and including various *Negotiators*, one for each negotiation that involves the participant at that given time. Negotiators either interface human beings with the negotiation framework through proper GUIs that allow him/her to place offers and counter offers, or they encapsulate

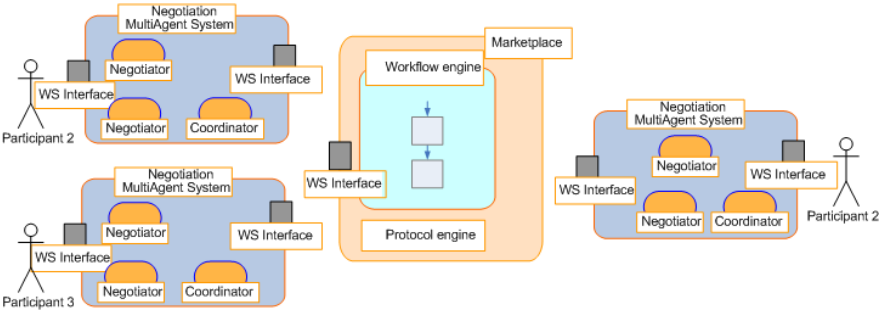

Fig. 1. The negotiation framework architecture

a decision model (see Section 4) that enables automatic negotiation to be executed. This allows us to support not only manual or automatic negotiations, but also hybrid negotiations where some participants are represented by automated agents, and some others are human beings.

Given that each negotiation participant has limited resources available, the result of one negotiation can impact on the participant ability to place an offer in another negotiation. For instance, if a telecom provider is negotiating with a consumer a high availability of its services, it might not be able to offer high availability to other consumers with which it has engaged other negotiations. In turn, if a consumer that is composing several services is accepting low level of performance from a service, it should be careful not to accept low level of performance from other services as well, otherwise the whole QoS of the resulting composition could become lower than required. In order to regulate these kinds of situations, each participant may exploit a *Negotiation Coordinator* that has the role of coordinating the action taken by the various Negotiators of the same participant. As regular Negotiators, the Negotiation Coordinator has a decision model that allows it to take decision at a higher level of abstraction.

The Marketplace defined in our framework is composed of two main parts, one taking care of the execution of the negotiation workflow and the second one controlling the correctness of the negotiation protocol. In particular, the Marketplace acts as an intermediary in all interactions among the participants, providing validity checks for the offers exchanged (through the *Protocol Engine*), based on their structure and the current state of the negotiation workflow. To make the search for agreements more efficient, the Marketplace is enhanced with a mediation function to guide the generation of the offers towards convergence of the individual objectives, based on the reactions of the participants. For example, in the one-to-one bargaining process whose implementation is described in Section 4, the mediator iteratively issues proposals to the parties. At each step, the given proposal is evaluated by the Negotiators, and if it is accepted by both, the negotiation ends successfully. Otherwise, a new proposal is generated based on the Negotiators evaluation. The mediator is implemented by an optimization algorithm, which will stop when no joint improvement is observed, i.e., at convergence to some offer, or if interrupted by the negotiation timeout.

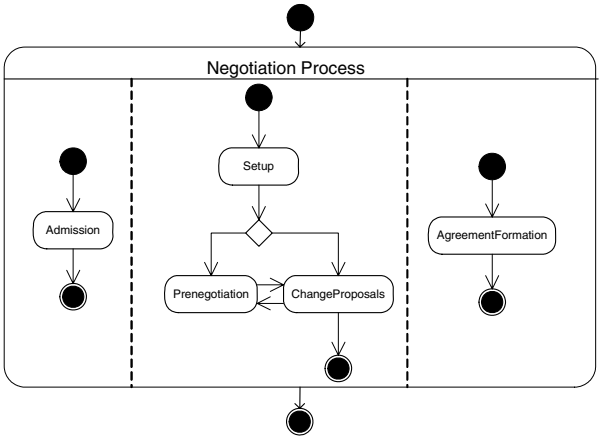

Fig. 2. The generic negotiation workflow

Our negotiation framework allows designers of negotiation to define the negotiation workflow as a Statechart using ArgoUML, and the negotiation protocol as a set of rules in the JBoss[1] Rule sintax. Figure 2 shows the Statechart associated to the most generic negotiation workflow. It can of course be replaced by more specific definitions. The framework, besides offering some predefined implementations of the decision model for Negotiators and Negotiator Coordinators, also allows the designer to define new decision models and to execute them. In the following sections we describe how the search-based optimization technique can be used to mediate one-to-one negotiation processes, and present an agent decision model.

4 Search-Based Negotiation Approach

As we have mentioned in the previous section, each Negotiator implements some decision model of the negotiation party, which can be arbitrarily configured beforehand. This usually implies to:

- define QoS attribute boundaries, expressed by constraints;
- identify the objectives to pursue, e.g., maximize one ore more utility functions expressed in terms of QoS attributes;
- prioritize the objectives and evaluate possible trade-offs among them;
- decide what information to make public, e.g., one of the above.

Negotiators may be equipped with a strategy, i.e., the algorithm to decide the reaction to a received offer at the given stage of the negotiation. In our approach, the strategy defines whether and how some of the above decision data, like the priorities of the attributes or the constraints, must change during the negotiation

[1] *http://www.jboss.org/*

at some pre-defined milestones. The idea is that low-priority attributes at the beginning of the negotiation may have their priority increased later, for example, prefer availability over the response time if the latter cannot be improved so far. Similarly, some constraints can be relaxed of some factor, representing the concession made by the Negotiator on some values for the attributes, to try to achieve a SLA when the timeout is about to expire.

Over a generated SLA proposal, each Negotiator reacts with a feedback. The feedback value for a proposal $o = (o_i)_{i=1,\ldots n}$ (n is the number of attributes and o_i the proposed value for attribute i), consists of a pair $(u = U(o,t), d = D(o,t))$, where $0 \leq u \leq 1$ represents the overall value (or degree of satisfaction) given to the proposal, and $0 \leq d \leq 1$ is a measure of the distance of the proposal from the acceptance region. These values may be computed as: $U(o,t) = (p_i(t) \cdot u_i(o_i))_{i=1,\ldots,n}$, where $u_i(o_i)$ is the utility value for the attribute i and $p_i(t)$ is the priority of the attribute at time t. Instead, given the constraint set at time t, represented as $cl_i(g,t) \leq 0,\ i = 1,\ldots,n$, the *distance from constraint satisfaction*, is:

$$D(g,t) = \sum_{i=1}^{n} cl_i(g,t) \cdot y_i, \tag{1}$$

where: $y_i = 0$ if $cl_i(g,t) \leq 0$ and $y_i = 1$ if $cl_i(g,t) > 0$.

In case of the one-to-one negotiation exploiting the mediation capabilities of the Marketplace (see Section 3), when the negotiation starts, the number of attributes and their domains are specified to identify the search space. Hence, according to the protocol in place, the mediation algorithm produces one or more proposals, to which fitness values will be attached. For the purpose of experimentations presented in this paper, the following fitness function (to be minimized) has been considered: $F(o,t) = eu \cdot (1 + e^{d1 \cdot d2})$, where, if $(u1, d1)$ and $(u2, d2)$ are the feedback values for offer o at time t received from the Negotiators 1 and 2, eu represents the Euclidean distance of $u1$ and $u2$. The rationale of this fitness is to equally accommodate the Negotiators preferences and to impose the offer to fall into the intersection area of the acceptance regions of the two Negotiators.

For the optimization problem, we propose to use meta-heuristic search algorithms, such as Hill-Climbing, Genetic Algorithms, and Simulated Annealing (SA). From the experiments we conducted, the latter outperformed the others, above all in terms of number of solutions required to converge. SA is a variant of the hill-climbing local search method (further details on these heuristics can be found in [8]). The SA approach constructed for the negotiation algorithm proposed in this paper works as follows: (i) it starts from a random solution; (ii) a neighbor solution of the current one is selected, by randomly choosing one QoS attribute and randomly changing its value within the admissible domain. The solution is then accepted if $p < m$, with: p a random number in the range $[0 \ldots 1]$, and $m = e^{\Delta fitness/T}$. The temperature T was chosen as:

$$T = T_{max} \cdot e^{-j \cdot r},$$

T_{max} being the maximum (starting) temperature, r the *cooling factor* and j the number of iterations. The process iterates until $T < T_{min}$.

5 Empirical Study

This section reports the empirical assessment of the search-based negotiation approach described in Section 4. In particular, the empirical study aims at answering the following research questions:

1. **RQ1:** To what extent the proposed negotiation approach is able to achieve feasible solutions for the different stakeholders?
2. **RQ2:** How do results vary for different QoS range overlaps?
3. **RQ3:** How do results vary for different utility functions?
4. **RQ4:** Are the performance of the proposed approach suitable for a run-time negotiation?

In the following we describe the experimentation context and setting, and then we report and discuss the obtained results.

5.1 Context and Settings

To assess the proposed negotiation approach, we set up a number of mediated one-to-one negotiation scenarios between a service provider and a consumer, bargaining over the average values of: price, response time and availability of a service. Although an SLA is usually concerned with ranges of values for each attribute, the empirical study focuses on negotiation of single values (representing the average, least, or maximum values), typical scenario that can be envisaged for the run-time binding of a service composition. Of course, our approach is also applicable to searching for agreements on QoS ranges, by specifying, in that case, the variability domains for the minimum and maximum values for each range or else the length of such ranges.

For these experiments, the global domain for the optimization algorithm was specified as in Table 1 (Domain column), where the price is expressed in Euro, the response time in seconds and the availability as a percentage. This domain may be agreeded by the Negotiators beforehand to limit the automatic generation of offers within realistic values, based on the type of service. We set up three different negotiation scenarios with different QoS acceptance sub-domains for the participating Negotiators, so to have sub domains of different negotiating agents — in particular of three providers negotiating with a consumer — overlapping by 80%, 50%, and 20%. An example of such a setting is given in Table 1.

Having fixed a negotiation timeout to a maximum 1200 generated offers, SA was configured as follows: $T_{max} = 0.30$, $T_{min} = 0.01$, $r = 0.0025$, Number of restarts = 3. For the fitness, we used the function $F(0, t)$ described in Section 4, which was reformulated in order to be maximized (e.g, replacing each constraint distance d_i with $1 - d_i$ and eu with $\sqrt{3} - eu$). As the focus of this empirical evaluation is on the effectiveness of the search-based approach, we considered

Table 1. Search domain and constraints of the negotiating agents

	Domain		Consumer		Provider_1		Provider_2		Provider_3	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
Price	0.10	5.50	0.10	3.40	0.76	4.06	1.75	5.05	2.74	5.50
Response Time	1.50	120.00	32.00	65.99	38.80	72.79	48.99	82.98	59.20	93.18
Availability	50.00	99.90	0.70	0.99	0.64	0.93	0.55	0.84	0.50	0.76

Table 2. Offers variability according to constraints overlapping percentage

	80% Overlap			50% Overlap			20% Overlap		
	Min	Max	Av	Min	Max	Av	Min	Max	Av
Price	1.63	1.66	1.64	1.75	1.77	1.76	2.74	2.76	2.74
Response Time	38.73	38.93	38.84	48.71	49.60	49.11	59.07	59.46	59.24
Availability	0.75	0.75	0.75	0.75	0.75	0.74	0.75	0.75	0.74

(i) the Negotiator's feedback functions as fixed during the negotiation, with a priority vector $[p_{price}, p_{rtime}, p_{availability}] = [0.4, 0.4, 0.2]$ for both providers and Consumer and (ii) fixed constraints. In this model, the feedback value of an offer from each Negotiator consists of a pair (u, d) where u is the vector of the attributes utility values, normalized in the interval [0,1] with respect to the overall domain of Table 1, and d is the normalized distance from constraints satisfaction. Experiments were performed on a Intel Core Duo T2500 2.0 GHz machine, with 1 GB of RAM. To avoid bias introduced by randomness, analyses were performed by replicating each run 10 times. Finally, a random search algorithm (i.e., offers are randomly generated) has been implemented in order to perform a sanity check of the SA-based approach.

5.2 Results

Question **RQ1** is concerned with the capability of the search-based approach to find one (or a set of) sub-optimal solution(s), according to the offer evaluating functions and constraints of the two Negotiators. Given the constraints setting of Table 1, the negotiation has been executed between Consumer and Provider_1, Consumer and Provider_2, and Consumer and Provider_3, using linear utility functions. Figure 3 reports, for each negotiation scenario, the fitness function evolution for SA (averaged over 10 runs), for each negotiation scenario. For these runs, we computed that 99% of the maximum fitness value was reached, respectively, after: 155, 137 and 200 generated offers in the worse case. The outcome of negotiations for different levels of overlap between the QoS admissible ranges of Provider and Consumer, is shown in Table 2. The final offers, although significantly different (according to the Kruskal-Wallis test, p-value=$2.5 \cdot 10^{-6}$), satisfy both negotiating agents' constraints and, when the overlap between Provider and Consumer domains decreases, the Consumer must expect higher cost and response times and lower availability (this answers to **RQ2**).

To answer **RQ3**, negotiation runs between Consumer and Provider_1 have been performed by using different utility function shapes, i.e., linear, exponential, and logarithmic. Minimum, maximum and average of the QoS attributes negotiated values with respect to the utility shapes are shown in Table 3.

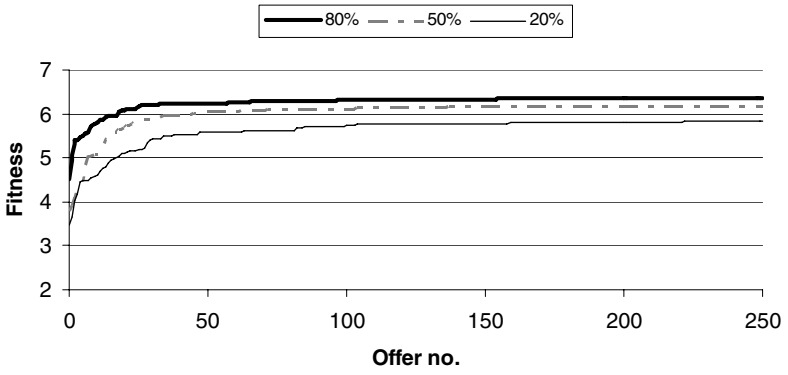

Fig. 3. Evolution of the negotiation for different overlap percentages of the QoS domains

Table 3. Offers variability according to utility shapes

	Linear			Exponential			Logarithmic		
	Min	**Max**	**Av**	**Min**	**Max**	**Av**	**Min**	**Max**	**Av**
Price	1.63	1.66	1.64	3.38	3.40	3.39	0.80	0.84	0.82
Response Time	38.73	38.93	38.84	40.03	65.49	55.24	38.74	39.00	38.85
Availability	0.75	0.75	0.75	0.78	0.78	0.78	0.73	0.734	0.732

In the exponential case, convergence is towards the boundary values of the Consumer's subdomain. Indeed, the Consumer's utility-based evaluations of the offers in the specified domain are much higher (for response time and cost, lower for availability) than those by the Provider (for response time, these could only be higher for values greater than the intersection point of the two utility curves, which is about of 119.30 s), and the Euclidean distance is minimal at the boundaries. Also, quite a high variation of the resulting response time can be observed across the different runs, as the normalized Consumer feedback values (which lead the fitness improvement as just explained) of offers of that region are all close to 1.0, thus they are all equally acceptable solutions.

Finally, to answer **RQ4**, we compared the performance of SA with that of a random-search (RS) algorithm. From all our experiments, it turned out that SA is both more efficient (in terms of best fitness reached) and faster. Also, we could observe that the difference increases for lower size acceptance sub-domains on the search space and/or low overlapping percentage. Figure 4 shows the outcomes of SA and RS when carrying out negotiations with a domain overlap of 20%. Although both SA and RS reach very similar values (SA final value is only 3% of that of RS), SA was able to converge significantly quicker to the 99% of the final value (p-value=0.0001 according to the Mann-Whitney test). Also, for random search, Negotiators' constraints were met only for the 30% of the runs, against 100% of SA. This indicates the soundness of choosing SA to drive the automatic, search-based SLA negotiation.

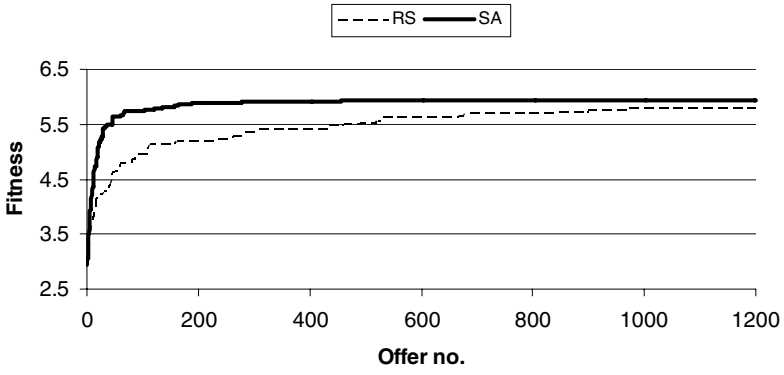

Fig. 4. Performance of SA vs Random Search (RS)

6 Related Work

Research works on automated negotiation are mainly related to architectural solutions for negotiation or algorithms and models for protocols and Negotiators strategies. In [3] a multi-agent framework with a two-layered architecture is presented, where local QoS negotiations for finding a binding to the same invoke activity are coordinated to satisfy global QoS constraints of a composition. Local constraints have to be inferred from the global ones and from the workflow topology. The model presented in [4] consists of a negotiation broker carrying out one-to-one negotiations on behalf of both service consumers and providers. The decision model of the negotiators is expressed by a hard-coded parametric function, which needs to be instantiated by the parties before negotiation starts. Instead, a marketplace-based architecture is presented in [12], where the marketplace mediates all the communication among negotiation parties, but it does not take part, itself, in the negotiations. In [9], the multi-agent system paradigm is combined with the web service technology to enable distributed online bargaining applications. However, only two negotiation processes are supported, bilateral and trilateral. The last uses a third entity to authenticate the trading agents and to validate the deals.

On the negotiation algorithms side, the existing approaches are generally concerned with two aspects: definition of decision models for the agents, and search for the near-optimal strategy, i.e., leading to Pareto optimal solutions. In [7] a strategy is defined as a weighted sum of tactics. Also, machine learning techniques can be used by the agents to learn decision rules from historical negotiation data ([10]). With respect to finding the optimal strategy, in [1], the Q-learning algorithm is used to select strategies as linear combinations of tactics, and convergence to optimality is proved to be reached after each action has been tried sufficiently often. Instead, in [11] Genetic Algorithms are used to evolve strategies whose fitness is computed according to their outcomes in negotiation runs. Finally, in [2], evolutionary methods are combined with cooperative

game theory to first explore possible agreements spaces and then to distribute the payoffs and find an optimized point.

In our work, we focus on the applicability of automated negotiation approaches to the web services world. We use heuristic-based optimization algorithms to try to speed up the process of finding possible agreements. Our approach can be integrated with other decision models and strategy evolution methods as part of the agents' implementations. Moreover, we present a strategy model, taylored for negotiation to support the dynamic binding of a composition, so that the single negotiation objectives can be tuned on the run, to try to obtain the best possible QoS at global level.

7 Conclusions

In this paper we have presented an architecture that supports the actuation of various negotiation processes and offers an efficient search-based algorithm to assist the negotiating parts in the achievement of an agreement. The interesting aspect of the architecture is the possibility of instantiating negotiation workflows and protocols defined by the designer as well as various decision models for Negotiators. As future work we plan to perform more experiments with different workflows, protocols (including also optimization strategies), and decision models to try to understand which of them is more interesting in the specific application domains we are considering in the SeCSE project. Also, we plan to experiment with real SLAs that will be provided by our industrial partners in the project.

Acknowledgments

This work is framed within the European Commission VI Framework IP Project SeCSE (Service Centric System Engineering) (*http://secse.eng.it*), Contract No. 511680.

References

1. Cardoso, H., Schaefer, M., Oliveira, E.: A Multi-agent System for Electronic Commerce including Adaptive Strategic Behaviours. In: Barahona, P., Alferes, J.J. (eds.) EPIA 1999. LNCS (LNAI), vol. 1695, pp. 252–266. Springer, Heidelberg (1999)
2. Chen, J.-H., Chao, K.-M., Godwin, N., Soo, V.-W.: A Multiple-Stage Cooperative Negotiation. In: EEE'04. Proc. International Conference on e-Technology, e-Commerce and e-Service, Taipei, Taiwan, pp. 131–138. EEE (March 2004)
3. Chhetri, M., Lin, J., Goh, S., Zhang, J., Kowalczyk, R., Yan, J.: A Coordinated Architecture for the Agent-based Service Level Agreement Negotiation of Web service Composition. In: ASWEC'06. Proc. of the Australian Software Engineering Conference, Washington, DC, USA, pp. 90–99. IEEE Computer Society Press, Los Alamitos (2006)

4. Comuzzi, M., Pernici, B.: An Architecture for Flexible Web Service QoS Negotiation. In: EDOC'05. Proc. of the Ninth IEEE International EDOC Enterprise Computing Conference, Washington, DC, USA, pp. 70–82. IEEE Computer Society Press, Los Alamitos (2005)
5. Deutsch, M.: Cooperation and competition. The Handbook of Conflict Resolution: Theory and Practice (22), 21–40 (2000)
6. Jennings, N., Faratin, P., Lomuscio, A., Parsons, S., Wooldridge, M., Sierra, C.: Automated Negotiation: Prospects Methods and Challenges. Group Decision and Negotiation 10(2), 199–215 (2001)
7. Matos, N., Sierra, C., Jennings, N.: Determining Successful Negotiation Strategies: An Evolutionary Approach. In: ICMAS 1998. Proc. 3rd International Conference on Multi-Agent Systems, Paris, FR, pp. 182–189. IEEE Press, Los Alamitos (1998)
8. Michalewicz, Z., Fogel, D.B.: How to Solve It: Modern Heuristics, 2nd edn. Springer, Berlin (2004)
9. Ncho, A., Aimeur, E.: Building a Multi-Agent System for Automatic Negotiation in Web Service Applications. In: Proc. of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, New York, pp. 1466–1467. IEEE Computer Society Press, Los Alamitos (2004)
10. Oliveira, E., Rocha, A.: Agents Advanced Features for Negotiation in Electronic Commerce and Virtual Organisations Formation Processes. In: Sierra, C., Dignum, F.P.M. (eds.) Agent Mediated Electronic Commerce. LNCS (LNAI), vol. 1991, pp. 78–97. Springer, Heidelberg (2001)
11. Oliver, J.: On Artificial Agents for Negotiation in Electronic Commerce. PhD thesis, Univ. of Pennsylvania (1996)
12. Rolli, D., Luckner, S., Momm, C., Weinhardt, C.: A Framework for Composing Electronic Marketplaces - From Market Structure to Service Implementation. In: WeB 2004. Proc. of the 3rd Workshop on e-Business, Washington, DC, USA (2004)