A Conceptual Framework for Automated Negotiation Systems

Manuel Resinas, Pablo Fernandez, and Rafael Corchuelo

ETS Ingenieria Informatica
Universidad de Sevilla, Spain
http://www.tdg-seville.info

Abstract. In the last years, much work have been done in the development of techniques for automated negotiation, and, particularly, in the automated negotiation of SLAs. However, there is no work that describes how to develop advanced software systems that are able to negotiate automatically in an open environment such as the Internet. In this work, we develop a conceptual framework for automated negotiations of SLAs that serves as a starting point by identifying the elements that must be supported in those software systems. In addition, based on that conceptual framework, we report on a set of properties for automated negotiation systems that may be used to compare different proposals.

1 Introduction

In this work, we focus on SLA negotiations[1]. The goal of this kind of negotiations is to reach an agreement between a service provider and a service consumer about the terms and guarantees of the service consumption. We are interested in developing software systems that are able to negotiate in open environments, such as the Internet, with temporal restrictions. That is, the agreement has to be reached in a limited amount of time. This scenario defines the characteristics of the negotiation that shall be carried out: it is a non-cooperative negotiation between two or more parties with partial information of the world and hard computational constraints.

In the last years, much work have been done on automated negotiations. These works are focused on the development of new decision-making algorithms or the construction of new protocols that presents certain desirable characteristics for automated negotiations. However, much less attention has been paid to the software artefacts that are necessary to carry out this automated negotiation. In this work, we analyse the problem of automated negotiation of service level agreements from a software engineering perspective. Specifically, we want to identify the elements that are required to build a system that develops an automated negotiation of SLAs.

To reach this goal, we present a novel conceptual framework of SLA automated negotiations. Unlike other works, we take a software engineering perspective and

[1] SLA negotiations are equivalent to the so-called service-oriented negotiations [1].

centre on the elements that are required to build an automated negotiation system. This conceptual framework settles the bases for a later analysis of the different negotiation proposals and defines a common vocabulary for automated negotiation systems. In addition, we obtain a set of properties of automated negotiation systems as a consequence of the conceptual framework. These properties may be used to evaluate and compare different proposals on automated negotiation of SLAs.

This paper is structured as follows: first, we describe the conceptual framework in Section 2, then, in Section 3, we use the conceptual framework to obtain a set of properties of automated negotiation systems. Finally, we present our conclusions and future work in Section 4.

2 A Conceptual Framework for Automated Negotiations

Traditionally, an automated negotiation system has been characterised by three elements [2]: protocol, negotiation object and decision-making model. In our conceptual framework we propose to extend this characterisation with two additional elements: information and preferences. Additionally, we detail more precisely the decision-making and protocol elements, and we also provide a concrete description of the negotiation object.

2.1 Negotiation Object

In a SLA negotiation, the object that is being negotiated is an agreement between parties. An agreement defines a dynamically-established and dynamically-managed relationship between parties [3]. The goal of the agreement is to establish the guarantees that must be observed during the execution of a service. An agreement is composed at least by the following:

- A specification of the *parties* involved in it. In principle, the number of parties involved in an agreement is not constrained. However, the most common case is two-party agreements.
- A collection of *terms* that describes both functional descriptions and non-functional guarantees of the service. Additionally a term can also express other aspects of an agreement such as termination clauses. A term is composed of three parts:
  - The *counterparty* whom the term is applied to. Each term is to be applied to one of the parties involved in the agreement and the party is obligated to fulfil what it is specified in it. Obviously, the *counterparty* must be one of those that have been designated in the agreement as one of the parties that are involved in it.
  - A set of *constraints* to specify functional or non-functional descriptions or guarantees of the service. It is expected that the content of these constraints will be very broad and domain-specific.
  - A set of *compensations* that will be applied in case the party does not fulfil the constraints specified in the term. This element is optional and it is not supported by the majority of the negotiation strategies.

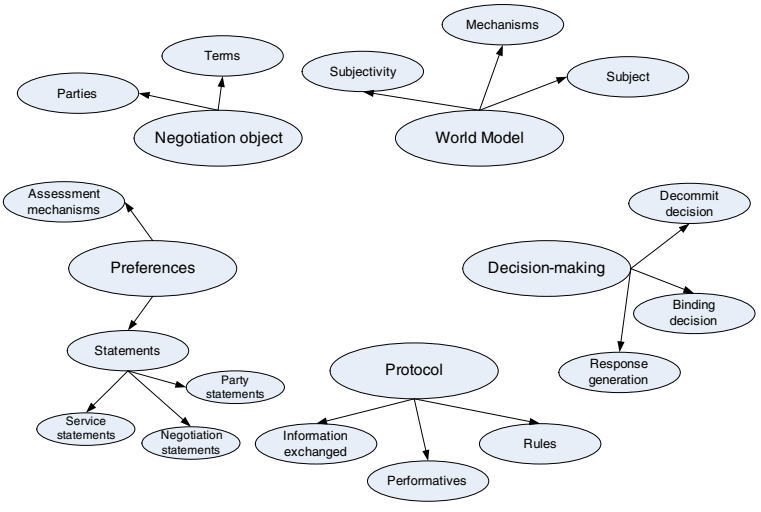

Fig. 1. Conceptual framework

2.2 Preferences

The agreement preferences express the data that is used to assure that the user needs are correctly dealt among the negotiation process. These preferences comprise: a set of *statements* expressing the *features* and *requirements* of the user, and an *assessment mechanism* to evaluate and compare agreement proposals. The most common way of evaluating proposals is through the definition of utility functions [4]. Statements can be classified depending on the domain in which they are applied:

- **Service statements**. These statements are applied to the service offered (or demanded). They can refer to either functional or non-functional characteristics of the service such as the *service interface* or the *service price*.
- **Party statements**. In this case, an expression about the party is stated. These statements can express either features or requirement over a given party. Examples of this can be: *Party Z is located in Iran* or *Party X must have a high reputation on service Y*.
- **Negotiation statements**. They specify features about the negotiation process itself, such as the negotiation deadline.

Each statement is linked to a set of *languages* that give semantics to the vocabulary used within the statement. Ontologies can be seen as an example of languages describing the relationship amongst concepts of a given semantic domain. Usually, the statements are expressed using two different formalisms: rules and constraints. However, other formalisms could be used.

2.3 World Model

While negotiating, the more information we have about other parties, the better our performance is [5]. Furthermore, it has been shown that taking the conditions of the market into account does improve the outcome of negotiations with several simultaneous opponents [6]. Therefore, it is essential for an automated negotiation system to build a model of the counterparties and the market. We call this model, the world model of an automated negotiation system and it may cover three different domains:

- Counterparties: including the characteristics of the service demanded or offered, the negotiation process followed by the counterparty and the counterparty itself (e.g. reputation or geographical location).
- Market: for instance, the market reservation price, or the probability of appearing outside options during the negotiation [7].
- Service-domain: such as knowledge about the vocabulary used in the terms of the agreement.

The information used to build the world model may be either objective or subjective: objective information typically includes the public features about the service demanded/supplied, but it may also include information about the counterparties themselves (e.g. their geographical location) and the market. Subjective information comprises elements such as the reputation of a counterparty or the market price of a certain service.

Finally, there are three different mechanisms to obtain the information that is used to build the world model: (i) Directly polling the potential counterparty. In this case, the system must implement a compatible specification of a format to express functional and non-functional features of services and a procedure to query and to inspect services. (ii) Querying a third party entity to obtain information related to a specific counterparty. For instance, to obtain information about its reputation or its geographical location. (iii) Analysing previous interactions with a potential counterparty. The results of the analysis may be stored in order to be used later while making decision about proposals related to the potential counterparty.

We envision that the first procedure shall be commonly used in gathering objective information about the counterparties, while the second and third procedures shall be more common in obtaining subjective information about the market and the potential counterparty itself.

2.4 Protocol

The negotiation protocol establishes the rules that govern the negotiation and the way the communication between the different parties involved in the negotiation is carried out as well as the information exchanged between the parties. We distinguish three different, although strongly related, aspects in a negotiation protocol: rules, performatives and information exchanged.

Performatives. A performative is the expression of the intention of the sender of a message. The set of performatives used in a negotiation protocol may differ significantly. However, there are a minimum subset of performatives that are common to the majority of negotiation protocols. Namely, *accept* (accept a proposal), *reject proposal* (withdraws the proposal), *reject negotiation* (cancel the whole negotiation process), and *commit*[2] (commit to a given proposal). Depending on the specific protocol that is being used, other performatives may be necessary. For instance, in auctions, it is common to use the *inform* performative to notify events occurred during the negotiation such as that a new bid has been done. In protocols that differentiates between binding and non-binding proposals, the *propose* (make a non-binding proposal) performative may be introduced. Other protocols use a vote system to decide which is the preferred offer [8]. In such protocols, a *vote* performative may be used. Finally, in negotiation protocols that use argumentation [9], other performatives to introduce arguments supporting our proposal are used such as *argue* and *challenge* [9].

Rules. In a negotiation protocol, there are usually some restrictions regarding a variety of aspects of the negotiation such as how the proposals must be built, when a participant can post a proposal, which performative can be used in each moment, when a participant may join to the negotiation, or when the negotiation can finish. In [10], a taxonomy of rules for negotiation protocols is presented. However, although it is a thorough taxonomy and the majority of the aspects of automated negotiation are covered, we believe that it should be extended with rules for decommitting from previously created agreements. We argue that there must be rules that explicitly specify whether a decommitment may take place, and, if so, when it may occur and which are the penalties to be paid as a compensation.

Information Exchanged. The third aspect in a negotiation protocol is the type of information exchanged amongst the participants in the negotiation. A variety of approaches has been presented in the literature. Those approaches may be classified into three broad groups:

- The information exchanged explicitly states the parts of the agreement that are disliked by the party as well as the proposed changes.
- The information exchanged consists only of proposals. In other words, the negotiation protocol is proposal-based. The advantage of this approach is that it unveils less information to the other parties. The disadvantage is that the lack of explicit information implies a blind search of a mutually acceptable agreement that may lead to longer negotiations and even to not to find any agreement at all.
- The information exchanged includes proposals, as in proposal-based protocols, and statements that are used to persuade or convince the opponent to

[2] In the literature, this performative is usually called *propose* meaning making a binding proposal. However, we prefer to leave the term *propose* to non-binding proposals and to use *commit* for binding proposals.

accept our proposal [11]. This approach is called argumentation and it is a promising field that may eventually overcome the drawbacks of the proposal-based negotiation [9]. However, the negotiators that support argumentation tend to be very complex and no argumentation approach has been applied to a real scenario yet.

2.5 Decision-Making

The decision-making model determines the way a party behaves while involved in a negotiation process. Three elements form part of the decision-making model of an automated negotiation system: the decision of what is considered an acceptable agreement and whether to commit to it, the construction of responses to this information, and the decision to decommit from a previously established agreement if possible.

Binding Decision. This decision includes determining when a binding proposal must be submitted and whether a binding proposal that has been received should be accepted. The binding decision depends on several factors that may vary depending on whether it is a service consumer who is making the decision or it is a service provider. Nevertheless, we can divide these factors into three broad groups:

- Preferences of the user. These preferences may be related to the contents of the agreement, the party we are negotiating the agreement and the negotiation process.
- The information the system have about the status of the market and other possible concurrent negotiations. For instance, we may be more reluctant to accept a proposal if we know it is very likely that in a short amount of time we will receive a proposal better than the current one [7].
- External factors that may prevent a party to commit to an agreement. For instance, the provider's capability to accept new agreements or the existence of dependencies amongst the agreements a service consumer wants to reach.

Together with the decision of making or accepting a binding proposal, it has to be decided when this decision take place. Usually, the decision is made as the proposals are received like in [12]. However, other approaches may be followed such as making the decision in some certain points in time previously defined.

Response Generation. Other important task in an automated negotiator is to decide which response must be sent to the other participants in the negotiation. On the one hand, this response is subordinated to the binding decision. On the other hand, the response generated must obey the rules imposed by the negotiation protocol. The process followed to generate the responses varies significantly. However, in general, it depends on the performatives of the negotiation protocol and the expressivity of the information exchanged during the negotiation:

- In auctions, the unique possible response is a bid together with the bidding price. Therefore, in this case the problem is centred on deciding in which auction must be placed the bid [13].
- In bilateral proposal-based protocols, a counterproposal must be generated. A wide variety of techniques have been developed to generate them. The most significant are: those that use time-dependant functions, resource-dependant functions, etcetera to obtain the counterproposal by modifying the values of the terms of the offer [4]; those that try to make the counterproposal more appealing to the opponent by sending the counteroffer with the highest similarity to the received offer [14]; those that use fuzzy constraint techniques [15], and those that interpret the negotiation as if it were a game and use techniques similar to those used in chess games [16]. Genetic algorithms have also been used to calculate offline which is the best strategy to use depending on the conditions of the negotiation in a certain instant [17].
- In negotiation protocols that supports argumentation, the response generation includes two problems [9]. First, the different arguments must be generated and then, the best argument from the point of view of the speaker must be selected. However, both the generation of arguments and the selection of the best one may occur at the same time. Nevertheless, whether or not this is possible depends on the specific argumentation framework.

Decommit Decision. It has been proved [18,12] that decommitment is a valuable resource when dealing with multiple simultaneous negotiations and, hence, it is another decision element that must be included in an automated negotiation system. The decommit decision is highly related to the binding decision and depends on the same factors. Therefore, in many cases, both the binding and decommit decisions are made by the same element [12].

3 Properties of Automated Negotiation Systems

The following properties for automated negotiation systems (ANS) have been obtained based on the conceptual framework (Section 2). We must remark that these properties are just centred on high-level details of automated negotiation systems and do not cover lower-level elements such as concrete technologies, protocols or algorithms.

1. *Information query*: An information query is an inquiry made by one party to another to obtain more detailed information about it or about the service it provides or demands.
2. *World model*: An ANS builds a world model if it analyses previous interactions with the elements external to the architecture and uses the results to make better decisions [2] during the negotiation.
3. *Third party information*: An ANS uses third party information if it explicitly queries a third party entity to obtain information related to another party.

4. *Information managed about the parties*: There are three types of information that can be managed: about the parties, the service and the negotiation process itself.
5. *Multiple negotiation protocols supported*: An ANS supports multiple negotiation protocols if it is able to use a number of negotiation protocols such as different bilateral negotiation protocols or several auction protocols [2].
6. *Decommitment from previously established agreements*: An ANS supports the decommitment [18] from previously established agreements if it can revoke previous agreements before the execution of the service.
7. *External factors in binding decisions*: An ANS may make use of a capacity estimator to determine whether the provider can provision a certain agreement before committing to an agreement [19].
8. *Cooperative or non-cooperative agreement creation*: An ANS supports non-cooperative agreement creation when it acts as a self-interested party reaching agreements with other self-interested parties. Alternatively, an ANS supports cooperative agreement creation when it can reach agreements with other parties trying to maximise the social welfare.
9. *Assessment mechanisms used*: The assessment mechanisms of an ANS is the kind of information used to evaluate the goodness of a proposal or agreement in relation to some criteria provided by the user.
10. *Forms of expressing preferences*: The preferences about the service and the parties can be expressed in different ways such as rules or constraints (see Section 2).

4 Conclusions

In this work, we analyse the problem of automated negotiation of service level agreements from a software engineering perspective. To develop advanced automated negotiation systems, it is necessary to identify the elements that are required in them. We present a conceptual framework for automated negotiation of SLAs that identifies the elements that must be supported by an advanced automated negotiation system. We extend the classical elements of automated negotiation with two new ones: world model and preferences. Furthermore, we detail more precisely the parts that compose the other three: negotiation object, protocol and decision-making. In addition, taking the conceptual framework as a basis, we also present a set of properties of automated negotiation systems that can be used to compare them.

Moreover, we believe that this work is a necessary step to develop a software framework for automated negotiation system that gives support to the most significant proposals of negotiation protocols and strategies that have been done in the last years.

Further work includes refining the categorisation of automated negotiation proposals in order to identify which alternative is better in each case. In this way, an automated selection of the algorithms used to carry out the negotiation can be made depending on the context of the negotiation.

Reference

1. Sierra, C., Faratin, P., Jennings, N.R.: A Service-Oriented Negotiation Model Between Autonomous Agents. In: Proc. of the 8th European Workshop On Modelling Auton. Agents in a Multi-Agent World, Springer-Verlag (1997) 17–35
2. Jennings, N.R., Faratin, P., Lomuscio, A.R., Parsons, S., Wooldridge, M., Sierra, C.: Automated Negotiation: Prospects, Methods and Challenges. Group Decision and Negotiation **10** (2001) 199–215
3. Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Pruyne, J., Rofrano, J., Tuecke, S., Xu, M.: WS-Agreement Specification (2004)
4. Faratin, P., Sierra, C., Jennings, N.R.: Negotiation Decision Functions For Autonomous Agents. Int. Journal of Robotics and Autonomous Systems **24** (1998) 159–182
5. Zeng, D., Sycara, K.: Bayesian Learning in Negotiation. Int. J. Hum.-Comput. Stud. **48** (1998) 125–141
6. Sim, K.M., Wong, E.: Toward market-driven agents for electronic auction. Systems, Man and Cybernetics, Part A, IEEE Transactions on **31** (2001) 474–484
7. Li, C., Giampapa, J., Sycara, K.: Bilateral negotiation decisions with uncertain dynamic outside options. In: Electronic Contracting, 2004. Proceedings. First IEEE International Workshop on. (2004) 54–61
8. Klein, M., Faratin, P., Sayama, H., Bar-Yam, Y.: Protocols For Negotiating Complex Contracts. IEEE Intelligent Systems **18** (2003) 32–38
9. Rahwan, I., Ramchurn, S.D., Jennings, N.R., McBurney, P., Parsons, S., Sonenberg, L.: Argumentation-Based Negotiation. The Knowledge Engineering Review **18** (2003) 343–375
10. Bartolini, C., Preist, C., Jennings, N.R.: Architecting For Reuse: A Software Framework For Automated Negotiation. In: Agent-Oriented Software Engineering III: AOSE 2002. Volume 2585 of Lecture Notes in Computer Science., Springer-Verlag (2003) 88–100
11. Ramchurn, S.D., Jennings, N.R., Sierra, C.: Persuasive Negotiation For Autonomous Agents: A Rhetorical Approach. In: Proc. IJCAI Workshop On Computational Models of Natural Argument. (2003) 9–17
12. Nguyen, T., Jennings, N.: Managing commitments in multiple concurrent negotiations. Electronic Commerce Research and Applications **4** (2005) 362–376
13. Anthony, P., Jennings, N.R.: Developing a Bidding Agent For Multiple Heterogeneous Auctions. ACM Trans. Internet Technology **3** (2003) 185–217
14. Faratin, P., Sierra, C., Jennings, N.R.: Using Similarity Criteria to Make Trade-Offs in Automated Negotiations. Artificial Intelligence **142** (2002) 205–237
15. Kowalczyk, R.: Fuzzy e-negotiation agents. Soft Computing **6** (2002) 337–347
16. Karp, A.H., Wu, R., Chen, K.Y., Zhang, A.: A Game Tree Strategy For Automated Negotiation. In: ACM Conference On Electronic Commerce. (2004) 228–229
17. Fatima, S.S., Wooldridge, M., Jennings, N.R.: A Comparative Study of Game Theoretic and Evolutionary Models of Bargaining For Software Agents. Artificial Intelligence Review **23** (2005) 187–205
18. Sandholm, T., Lesser, V.: Leveled commitment contracts and strategic breach. Games and Economic Behavior **35** (2001) 212–270
19. Ludwig, H., Gimpel, H., Dan, A., Kearney, R.: Template-Based Automated Service Provisioning - Supporting the Agreement-Driven Service Life-Cycle. In: ICSOC. (2005) 283–295