A Software Framework for Automated Negotiations of SLAs

Manuel Resinas, Pablo Fernandez and Rafael Corchuelo

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

To make automated negotiation widely used in real scenarios, it is necessary to develop advanced software systems that are able to carry out such negotiation in those scenarios. However, although much work has been done in automated negotiation, most of these efforts have been focused on the development of negotiation protocols and strategies and only a few are centred on software frameworks for automated negotiation. In this paper, we detail a software framework to develop automated negotiators of service level agreements (SLAs). Unlike other proposals, we provide a detailed description of elements for full support of multiple negotiation protocols, building complex models of the world, and advanced decision-making including strategy selection and creating counterproposals.

Introduction

In this work, we focus on SLA negotiations\(^1\). The goal of this kind of negotiation is to reach an agreement between a service provider and a service consumer about the terms and guarantees of the service consumption. From now on, when we use the term automated negotiation, we are referring exclusively to service agreement automated negotiations. In the last years, much work has been done on automated negotiation. These works are focused on the development of new decision-making algorithms or the construction of new protocols that present 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 paper, we report on a software framework to support automated negotiations of SLAs. The novelty of our proposal lies in two characteristics. First, we use a service-oriented approach to define the elements (services and data) of the framework. Second, unlike other proposals, we cover all required elements that are necessary to create a complex automated negotiation system [6] including world modelling, decision-making, preferences, negotiation protocols and negotiation object. The structure of this paper is the following. First, in Section 2 we detail the software framework. Next, we report on related work in Section 3, and we conclude in Section 4.

\(^1\) Also known as service-oriented negotiations [7]
A software framework for automated negotiations of SLAs

In this work, we focus on the particular case of concurrent proposal-based bilateral negotiations of service agreements. The rationale is that proposal-based bilateral negotiations are a very well studied field. Therefore, it is mature enough to start developing software systems that work in a real scenario. Furthermore, they are aligned with recent efforts of the industry in this area such as WS-Agreement [1].

The goal of the proposed software framework is to provide engineering support for software developers to develop an automated negotiation system that implements the conceptual framework for automated negotiations described in [6]. Therefore, the framework must provide the following: (i) data structures to represent a service agreement and the user preferences that guide the negotiation process; (ii) services to manage several negotiation protocols; (iii) services to support the decision-making, and (iv) services to create models of other counterparties and the market.

The software framework developed uses a service-oriented based approach. Figure 1 depicts an overview of the services\footnote{Note that not all relations are depicted in the diagram.} that compose the framework and their relationships. The framework provides a specification for each service and a reference implementation for two of them: bilateral negotiator and bargaining coordinator. Next, we detail the software framework by analysing how the elements of the conceptual framework are supported.

- **Service agreement data structure.** The agreement is composed of the parties involved in it and a set of terms that define its content. These terms comprise the counterparty to which it is applied, constraints to specify functional or non-functional descriptions or guarantees of the service, and, optionally, compensations that must be paid if the term is not fulfilled.
• **Preferences data structure.** The agreement preferences are composed of a set of statements that express features and requirements about a service, an assessment mechanism that is used to evaluate the received proposals and it is usually defined by means of utility functions, and a set of statements expressing the desired characteristics of the negotiation process.

• **World modelling services.** The goal of these services is to manage the world model that is necessary for the decision-making services to make more accurate decisions. There are five types of world modelling services: *Inquirer and informant services* that are in charge of obtaining and publishing the public information of a service respectively; *estimator services* that handle the subjective information about other participants, the market and the service domain; *estimators repository services* that are responsible for returning endpoints to the specific estimator services, and *archiver services* that serve as a storage of the history of all negotiations that have been developed by the system.

• **Protocol management.** The software framework gives support to the protocol element defined in the conceptual framework as follows: (i) A generic proposal data structure, and (ii) a *protocol enforcer service* that acts as a proxy between the negotiator service and the counterparty, transforms the concrete protocol messages into the proposal data structure defined above and vice versa, and it ensures that the protocol rules are obeyed.

• **Decision-making services.** The decision-making services defined by the software framework provide mechanisms to support the binding decision and response generation. The binding decision is supported by (i) the *commit handler service*, which has the final decision on when a binding proposal must be submitted and whether a binding proposal that has been received should be accepted, and (ii) *commit advisor services*, which advise about the convenience to commit to a given proposal. The response generation services can be divided into negotiation services and coordination services. The negotiation services are: (i) *bilateral negotiator service*, which coordinates all other negotiation services in order to carry out a proposal-based bilateral negotiator with one unique counterparty; (ii) the *response selector service*, which chooses the response that shall be sent to the counterparty; (iii) *strategy services* that build a counterproposal given the current proposal submitted by the counterparty, and (iv) the *strategy selector service* that decides the most appropriate strategy service to use. The coordination services include: (i) a *bargaining coordinator service* that coordinates all simultaneous negotiations and acts as a bridge between them and the binding services, and (ii) *policy generator services* that define the policies that are sent to each negotiator depending on the status of all concurrent negotiations.

**Related work**

Most related work is complementary to ours because it is focused on the negotiation object, protocols and decision-making models and, thus, they can be used to implement the services specified in our framework. For instance, WS-Agreement [1] may be used as the concrete format of the negotiation object and its templates
mechanism can be used as a way for the informant service to publish preferences. Decision-making is usually developed by negotiation strategies that are algorithms that are used to generate counterproposals based on the last proposal received [3,4].

Closer to our work is [2] that presents a software framework for automated negotiation. However, this framework is only concerned with negotiation protocols. There are other software frameworks for automated negotiation [8, 5] but they do not support direct information query, or third party information. They also do not support a change of strategy during the negotiation. Finally, subjective information is not managed [8] or if managed, it is vaguely specified and do not cover how it is updated.

Conclusions

In this paper, we present a software framework for concurrent proposal-based bilateral negotiations of service agreements. The software framework proposed specifies different types of services that give support to the five elements of the conceptual framework: negotiation object, preferences, world modelling, protocol and decision-making. Unlike other proposals, our framework offers full support for building complex world models that may be required during an automated negotiation. In addition, we give support for developing an advanced binding decision and for changing dynamically of negotiation strategy depending on the negotiation context.

References