PROCEEDINGS OF THE WORKSHOP ON APPLICATIONS OF SOFTWARE AGENTS
ISBN 978-86-7031-188-6, pp. 25 - 31, 2011

Experiments with Protocols for Service Negotiation*

Costin Bădică and Mihnea Scafeş

University of Craiova, Software Engineering Department
Bvd.Decebal 107, Craiova, RO-200440, Romania
{badica_costin scafes_mihnea}@software.ucv.ro

Abstract. In this paper we provide experimental results concerning the impact of the negotiation protocol onto the quality of the negotiation outcome as well as onto the communication complexity of interactions incurred during negotiations. We evaluate experimentally three negotiation protocols (Direct Task Assignment, Contract Net and Iterated Contract Net) with respect to two performance measures: negotiation outcome (i.e. utility) and communication complexity (i.e. number of messages transferred), by assigning different busy profiles to the contractors. We find that the Direct Task Assignment delivers the worst average outcome, but at the same time it uses the lowest number of messages. The Contract Net and Iterated Contract Net deliver much higher utility on average, but the Iterated Contract Net obtains the highest outcome for some configurations at the cost of the highest number of messages.

1 Introduction

We have developed a conceptual framework for service negotiation that addresses protocols, subjects and decision components in a collaboration system for helping human experts and population to deal with disasters (see the FP7 DIADEM project[1] that targets crisis management in the context of chemical incidents in industrial and urban areas). Our framework supports generic one-to-many negotiations and it defines two roles: manager and contractor [4, 2]. The manager is the agent that requests a service and thus initiates the negotiation. The contractor is the agent that is able to provide the service requested by the manager. For a more complete review of the conceptual framework, please see [1]. A brief description of the design and implementation is given in [3].

Currently we have configured our framework with three negotiation protocols that we have found useful in disaster and environment management problems. These protocols are *Direct Task Assignment (DTA)*, *Contract Net (CNET)* and *Iterated Contract Net (ICNET)*[2]. For more information see [3].

Negotiation participants playing either the manager or contractor roles use utility functions to quantify their preferences over proposals. In our framework the manager

*Mihnea Scafeş was supported by IOSUD-AMPOSDRU contract 109/25.09.2008.

[1]DIADEM Distributed information acquisition and decision making for environmental management: http://www.ist-diadem.eu/.

[2]*CNET* and *ICNET* are standardized by Foundation for Intelligent Physical Agents, see http://www.fipa.org/specs/fipa00029/ and http://www.fipa.org/specs/fipa00030/

uses a weighted additive utility function over the negotiation issues to evaluate proposals and to select the service provider.

2 Experiments

Let us assume that a manager agent M is negotiating for contracting a service from a contractor agent that is member of a set of n contractors $C_1, \ldots, C_n$. Each contractor C_i is characterized by her profile defined as a triple $([u_i^{min}, u_i^{max}], c_i, b_i)$. Each contractor C_i can offer a utility value u_i to the manager such that $u_i \in [u_i^{min}, u_i^{max}] \subseteq [0, 1]$. $c_i \in [0, 1]$ is the probability that she will be able to satisfy the requirements set by the manager's request. C_i has a busy profile $b_i = \{b_{i_1}, b_{i_2}, \ldots b_{i_m}\}$, where m is the maximum number of iterations and b_{i_j}, $1 \leq j \leq m$ is the probability of the contractor C_i being busy during iteration j. We assume that always $b_{i_m} = 0$, so that C_i will propose during the last negotiation iteration. The busy profiles are not taken into account when using the *DTA* and when using *CNET*, only the busy probability of the first iteration is taken into account. Busy profiles are mostly taken into account when using the *ICNET* protocol, which is based on multiple iterations. Depending on the utilized protocol the negotiation will incur a certain communication cost estimated as the number of messages exchanged between the manager and the contractor during the negotiation. Moreover the quality of the negotiation outcome will be estimated as the utility perceived by the manager for the contracted service.

If the manager is utilizing the *DTA* negotiation protocol then she will randomly assign the task to one of the contractors. However, a contractor that does not meet the requirements will not be able to provide the service, so she will have to report failure. In this case the manager will randomly select another contractor and so on, until a suitable contractor is found (we assume this is always the case). Note however that this trial-and-error process performed by the manager affects the outcome of the negotiation by decrementing her perceived utility. More precisely, if the successful contractor C_i that could perform the task was selected in the k-th trial then the utility perceived by the manager will be $u_i \times (1 - (k-1)/n)$ rather than u_i. Moreover, the communication cost associated to this negotiation interaction consists of $2 * k$ message exchanges.

If the manager is utilizing the *CNET* negotiation protocol then she will select the contractor C_i that provides her the highest utility u_i from those contractors that met the requirements of the call for proposals and were not busy (i.e. they were able to bid). The communication cost consumed for a busy contractor consists of 2 message exchanges, while for a not busy contractor (it doesn't matter if she could met or not the requirements, according to the *CNET* negotiation protocol [4] she either proposed or refused to bid) the communication cost consists of 3 message exchanges.

If the manager is utilizing the *ICNET* protocol we assume that she will perform as many negotiation iterations as needed to select one contractor. We simplify the negotiation by assuming that contractors do not change their bids between iterations. This assumption is not as restrictive as it might look, because some contractors are busy and can bid only in a late iteration. Moreover, we also assume that a busy contractor that meets the requirements of the call for proposals will always find time to bid during a certain negotiation iteration. The communication cost for a busy contractor consists in 2 message exchanges, while for a non-busy contractor it consists of 3 message exchanges, for each iteration.

In the simulation we considered one manager and 100 contractors.

There are at most 3 negotiation iterations and we consider 3 busy profiles that we can assign to contractors:

1. available during the first and the second iteration with equal probability. In this case, $b_i = \{0.5, 0, 0$. This profile says that a contractor will be able to respond during the first or the second iteration, but no later than the second iteration. These contractors respond during the first stages of the negotiation.
2. mostly busy during the first iteration, but available during the second iteration for sure. In this case, $b_i = \{0.9, 0, 0$. This profile says that a contractor will be able to respond during the first iteration in few situations, but it will surely respond during the second iteration. These contractors respond during the middle stage of the negotiation.
3. not available during the first iteration, but available during the second and third with equal probability. In this case, $b_i = \{1, 0.5, 0$. Contractors having this profile do not respond during the first iteration, but they will respond during the second or the third iteration, no later than the third iteration. These contractors respond during the last stages of the negotiation.

In the experiment, we assign the three busy profiles to contractors by following a certain configuration and for each configuration we run a bundle of 2000 negotiations. We vary the percentage of contractors that have been assigned the three busy profiles with a step of 5%. For example, we start from $(0, 0, 100)$, meaning that all the contractors have been assigned the third busy profile and then we continue with $(0, 5, 95), (0, 10, 90), \ldots (100, 0, 0)$, in the final configuration all the contractors having been assigned the first busy profile.

The values of the utilities u_i are randomly selected for each negotiation instance assuming uniform distributions. For this experiment, we consider $[u_i^{min}, u_i^{max}] = [0, 1]$. The status of the contractor (as satisfying or not satisfying the requirements of the manager) is randomly selected for each negotiation instance, according to the probability c_i. We fixed the probability c_i to 0.5 for all negotiations, for all configurations and for all contractors.

We are interested in how each of the studied protocols performs in terms of outcome and message traffic for each configuration of contractors. We run the negotiations in the experiment for each negotiation protocol.

Figure 1 shows the utility of the manager when using the *DTA* protocol. The scale labelled "first" shows the percentage of the contractors that can propose mostly during the first iterations of the negotiation, i.e. they have been assigned the first busy profile. The scale labelled "middle" shows the percentage of contractors that have been assigned the second busy profile. The percentage of the contractors that have been assigned the third profile is not shown in the figure, but it can be obtained by taking into account the fact that the sum of all the percentages is 100.

The maximum utility for *DTA* is a little over 0.5, making this the most inefficient protocol in terms of utility. The status of being "busy" is not taken into account by the

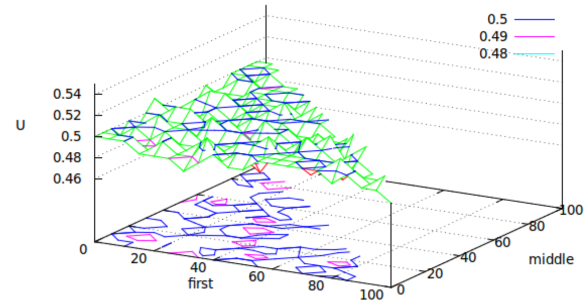

Fig. 1. Utility for DTA

manager when she is playing the *DTA* negotiation protocol, i.e. if she selects a certain contractor then the task will be assigned to her in any case. Nevertheless, if the assigned contractor cannot finalize the task successfully then she will report failure and consequently the manager will retry the operation of service contracting by assigning the task to another contractor. However, the utility is almost the same for all configurations, the average being around 0.5.

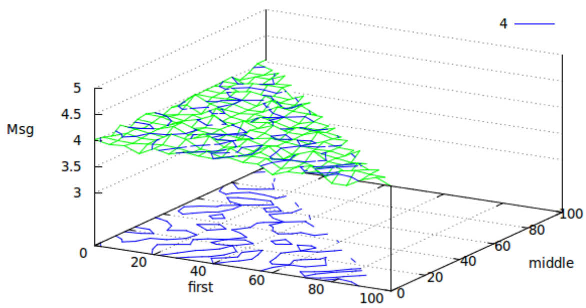

Fig. 2. Messages for DTA

Figure 2 shows the message statistics for the same protocol. In terms of messages, *DTA* performs very well, being the protocol with the lowest number of messages, outperforming the other protocols by far.

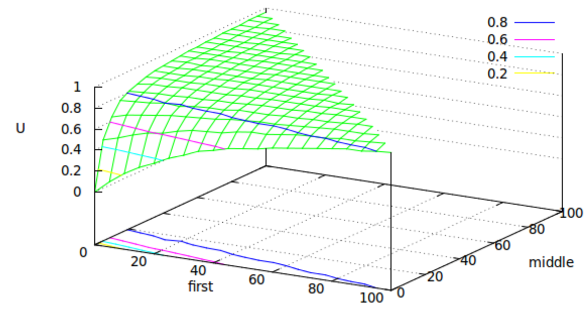

Fig. 3. Utility for CNET

Figure 3 shows the utility for *CNET*, which was expected to decrease as the number of contractors that propose during the first iteration decreases, because it receives less proposals and the probability to receive good proposals (i.e. of high utility) decreases. An interesting fact is that the utility decreases almost exponentially as the number of contractors that propose during the first iteration decreases.

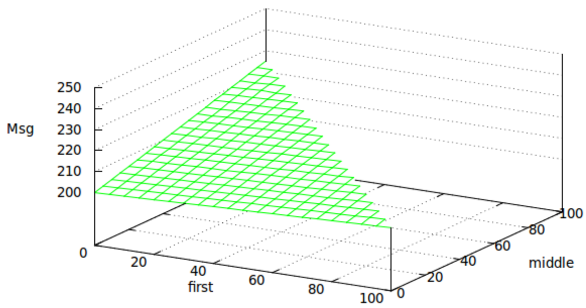

Fig. 4. Messages for CNET

The message count increases linearly with the number of contractors that propose during the first iteration (Figure 4).

For *ICNET* (Figure 5), the utility decreases almost as for *CNET* until a point where almost all contractors propose during iterations 2 & 3. Then it grows back to the initial maximum very fast.

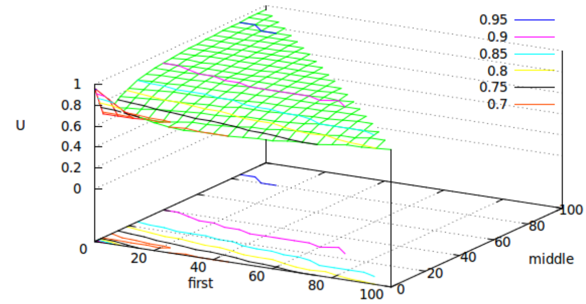

Fig. 5. Utility for ICNET

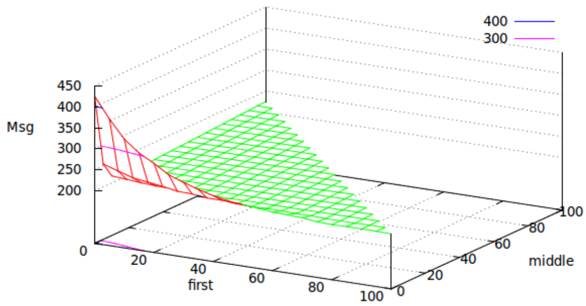

Fig. 6. Messages for ICNET

When most of the contractors propose during iterations 2 & 3, the message count grows exponentially to over 400 (see Figure 6. For the rest of the configurations, the message count varies almost the same as for *CNET*.

As future work we plan to expand the experiments in at least two directions: (i) to consider more complex negotiation instances that take into account more negotiation iterations, as well as that contractors might change their bids and managers can change their strategy for accepting contractors' bids during each iteration; (ii) to consider more complex workflows involving at least two interdependent negotiations such that the contracted service might also involve contracting of other required services.

References

1. Bădică, C., Scafeş, M.: Conceptual Framework for Design of Service Negotiation in Disaster Management Applications. In: Bai, Q., Fukuta, N. (eds.), Advances in Practical Multi-Agent Systems, Studies in Computational Intelligence 325, 359–375 Springer Verlag (2010)
2. Paurobally, S., Tamma, V., and Wooldridge, M.: A Framework for Web service negotiation. ACM Transactions on Autonomous and Adaptive Systems 2(4), ACM Press (2007)
3. Scafeş, M., Bădică, C., Pavlin, G., and Kamermans, M.: Design and Implementation of a Service Negotiation Framework for Collaborative Disaster Management Applications. Proceedings of the 2nd International Conference on Intelligent Networking and Collaborative Systems INCOS'2010, 519–524 (2010)
4. Smith, R.G.: The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. IEEE Transactions on Computers 29(12), 1104–1113 IEEE Computer Society (1980)
5. Yang, J., Li, W.-L., and Hong, C.-Y.: An Improvement to CNCP in Large-Scale Muti-Agent System. Proccedings of the 3rd International Conference on Innovative Computing Information and Control ICICIC'08, 147 (2008)