An Interaction Language for Argumentation-based Negotiation

Sonia V. Rueda María Vanina Martínez [1]

Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (LIDIA) [2]
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur. Av. Alem 1253, (8000) Bahía Blanca, Argentina
Tel: ++54 291 4595135 - Fax: ++54 291 4595136
{svr,mvm}@cs.uns.edu.ar

Abstract

Language is the medium used by heterogeneous and autonomous members of a community to communicate and to exchange knowledge. Social work allows members in a community to resolve problems that they could not face individually, but they must be able to *interact* in order to do it. Furthermore, is necessary to have a negotiation mechanism that allows members of a system to interact through it, trying to reach an agreement in order to satisfy their goals.

Our work is oriented towards the design of an interaction language that allows argumentation-based negotiation among collaborative BDI agents.

1 Introduction

Language is the medium used by members of a community to communicate and to exchange knowledge. The social complexity of a community is strongly bound to the language's evolution level, which can promote or obstruct some sorts of interaction among its members. The language's role is essential in the development of software applications based on the agents model. In this kind of application we can distinguish between the agents' *specification languages* and their *communication language*. The specification of the latter needs two essential elements, which are the *interaction language* and the *interaction protocol*.

In this work we describe a research line oriented towards designing an interaction language in order to allow *argumentation-based negotiation* among BDI agents. Negotiation is the basic mechanism of interaction because it allows agents to interact through it, trying to influence the plans or preferences of the rest of the members, until they reach an agreement.

In our model, the negotiation process begins when an agent's knowledge is not enough to reach its intentions and it requests collaboration from other members in the group by means

[1]Partially supported by *Secretaría de Ciencia y Tecnologa, Universidad Nacional del Sur*, CONICET, and by the *Agencia Nacional de Promoción Científica y Tecnológica* (PICT 202 Nro 13096 and PICT 2003 15043)
[2]Member of the IICyTI (Instituto de Investigación Científica en Ciencia y Tecnología Informática).

of a general request. Some of the group's members will be commited to their own goals and will not be able to collaborate. Others will respond instead by showing their willingness. The agent which performed this request starts a dialogue through a specific collaboration proposal with one of the agents that are willing to help.

If the negotiation process ends successfully, the *shared knowledge* between the two agents which have participated is modified. In this proposal, each agent's knowledge will depend not only on its perception capability but also on the outcomes of a dialogue. The individual knowledge of each agent will be conformed by its own specific knowledge and the knowledge shared with the other members in the group. The shared knowledge is distributed among pairs of agents [8], and the negotiation process therefore does not demand the agreement among all the members of the system. The individual knowledge provides the agent with a partial but consistent view of the world.

2 Dialogues and Conversations

When an agent, a, receives a message from another agent, b, requesting collaboration, it attempts to build a plan which must not be in conflict with its own intentions. If the agent's knowledge is not enough to make a plan, it can continue the dialogue with a counter-proposal. If agent a cannot attend this counter-proposal, agent b can make the request to another agent in the group (agent c) establishing a new dialogue this time between agent b and agent c. The dialogue between the two agents continues until an agreement is reached, or one of them decides that it is not able to collaborate. This last situation could arise due to lack of knowledge, or conflicts among goals.

If there exists a conflict between agent a and agent b, the former can repeat its request, but this time with more vehemence. The language must offer primitives that allow agents to express the level of demand imposed in the dialogue, as shown in [8].

The existence of conflicts between agents b and c can also arise due to a request performed by the former in order to attend an initial request from agent a. In these cases, our negotiation model proposes that agent b should inform agent a that the conflict with agent c is preventing it from collaborating; consequently agent a must be the one which insists agent c on resolving the conflict and reaching an agreement. Once again, the language must include suitable primitives that allow to express each agent's position in the dialogue.

In our work, dialogues are always circumscribed to pairs of agents; therefore, in each negotiation primitive the two first agents that appear are the sender and the receiver of the message, and the dialogue is involving only them. Starting from a single dialogue, the negotiation process can derive into a set of dialogues among other pairs of agents; we use the term *conversation* to represent these sets of dialogues. This last consideration extends the proposal in [7], where agents establish a dialogue with another agent and the negotiation process consists of a single dialogue.

During the whole conversation no agent executes actions because the modification of the knowledge could affect the plans that were elaborated. Thus, the language must allow agents to manifest not only their expectation that another member of the group remains committed to the negotiation, but also their intention to free it from that commitment.

Our model proposes that agents's collaboration requests are restricted only to requests for other agents' beliefs, and the possibility of requesting the execution of an action is not considered, at least directly through a primitive. When an agent requests collaboration for p it is indicating that it needs to include p in its own knowledge, and another agent has to add it on his behalf.

3 An Interaction Language

A social agent's reasoning process is substantially more complex than that of an isolated agent. Each agent will act motivated by its own goals and conditioned not only by its own beliefs, but also by those of other members. The existence of other members will allow each agent to request and obtain collaboration, but some sort of interaction capability will be necessary.

The role of a language for BDI agents is fundamental in allowing them to express their mental attitudes. The purpose of an interaction language, as is prposes in [4] is the communication of messages, which represent the agent's knowledge and that are interpreted in a well defined manner. Also, these messages cause certain actions on behalf of both the sender and receiver.

The interaction languages based on *Speech Act Theory* [9] capture the essential characteristics of human communication and translate them into a model suitable for the development of artificial agents. The key idea is the recognition of internal aspects of an autonomous artificial entity, taking special consideration in the changes produced due to the interaction with others. An interaction language among agents, based on *Speech Act Theory*, is constituted by a set of communication primitives like the one we show below:

- Primitives to start the negotiation process
  - `Req_col(`$\mathbf{a}_1$`).` Agent $\mathbf{a}_1$ asks the rest of the members in the system which of them are available for consider a collaboration request.
  - `Disp(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`).` Agent $\mathbf{a}_1$ lets agent $\mathbf{a}_2$ know that it is available for considering its collaboration request.
- Collaboration Requests
  - `Request(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`, `$\mathbf{a}_3$`, `p`, `Q`).` Agent $\mathbf{a}_1$ needs p and requests $\mathbf{a}_2$'s collaboration in order to obtain it. Q is the set of beliefs that must be avoided in the plan to obtain p.
  - `Insist(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`, `$\mathbf{a}_3$`, `p`).` Agent $\mathbf{a}_1$ asks agent $\mathbf{a}_2$ to revise its plans in order to avoid any conflict with p.
  - `Demand(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`, `$\mathbf{a}_3$`, `p`).` Agent $\mathbf{a}_1$ asks agent $\mathbf{a}_2$ to revise its goals in order to avoid any conflict with p.

- `Free(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`, `p`)`. Agent $\mathbf{a}_1$ indicates to $\mathbf{a}_2$ that it is free from the previous request for p.
- `Still_Int(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`, `p`)`. Agent $\mathbf{a}_1$ lets $\mathbf{a}_2$ know that it is still interested in agent $\mathbf{a}_2$'s help in obtaining p. Thus, agent $\mathbf{a}_2$ is committed to reserve its availability for obtaining p and to not change anything in its individual knowledge that could prevent it from obtaining p.

The first three primitives have a third optional argument that, when instantiated, indicates that the request is not being done directly by $\mathbf{a}_1$, but it is requesting p to $\mathbf{a}_2$ on behalf of $\mathbf{a}_3$. These extended versions of the primitives can be used when a conflict between agent $\mathbf{a}_2$ and agent $\mathbf{a}_3$ is preventing $\mathbf{a}_3$ from accepting a request from agent $\mathbf{a}_1$. In these cases, agent $\mathbf{a}_1$ must do the request to $\mathbf{a}_2$ informing him that the request is on behalf of $\mathbf{a}_3$.

- Responses to a collaboration request
  - `Accept(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`, `p`)`. Agent $\mathbf{a}_1$ informs agent $\mathbf{a}_2$ that it is able to collaborate because it knows p or because it can build a plan for it. Furtheremore there are no conflicts in adding p to the shared knowledge.
  - `Unable(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`, `p`)`. Agent $\mathbf{a}_1$ informs agent $\mathbf{a}_2$ that it is not capable of collaborating in obtaining p.
  - `Reject(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`, `p`)`. Agent $\mathbf{a}_1$ informs agent $\mathbf{a}_2$ that there exists a conflict between its own plans and p.
  - `Indirect_Reject(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`, `$\mathbf{a}_3$`, `p`, `Q`)`. Agent $\mathbf{a}_1$ informs agent $\mathbf{a}_2$ that it needs the beliefs contained in set Q in order to obtain p, but there exists a conflict between these beliefs and agent $\mathbf{a}_3$'s individual knowledge.
  - `Done(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`, `p`)`. Agent $\mathbf{a}_1$ informs agent $\mathbf{a}_2$ that it has performed all the necessary actions in order to obtain p and that it has been added to the shared knowledge.
- Requests for the modification of shared knowledge
  - `Request_add(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`, `p`)`. Agent $\mathbf{a}_1$ asks agent $\mathbf{a}_2$ for authorization to add p to their shared knowledge.
  - `Authorize_add(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`, `p`)`. Agent $\mathbf{a}_1$ responds to agent $\mathbf{a}_2$'s request indicating that the addition of p to their shared knowledge does not cause inconsistencies with its own beliefs nor with its own goals.
  - `Reject_add(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`, `p`)`. Agent $\mathbf{a}_1$ rejects agent $\mathbf{a}_2$'s request for the addition of p to the shared knowledge due to a conflict with its own beliefs or goals.

The request primitives do not in themselves cause the addition of the belief to the global knowledge. The actual addition will be performed once an agreement has been reached.

4 Conclusions

Agents metaphor suggests a model for the development of software with a stress on purpose, autonomy, reactivity, adaptability, and cooperation. A software entity having these characteristics can act in a suitable way on heterogeneous and distributed environments. Social work

allows them to resolve problems that they could not face individually, but in order to do it they must be able to interact. Interaction generates a negotiation process consisting of a sequence of messages exchanged among agents trying to reach a convergence of interests, which are initially divergent. If the process is successful, the interaction ends when a shared agreement is reached.

We developed a set of primitives which conform the interaction language for a system of collaborative BDI agents, which use argumentation-based negotiation in order to reach an agreement. Furthermore, a set of examples has been analyzed in order to determine most of the possible situations that could arise in a typical negotiation. This analysis was then used to show that our model is general enough to handle these situations. This research line is complemented with work focused on the description of an interaction protocol for the model and finding a suitable way to represent it.

References

[1] Bratman, M., Israel, D., and Pollack, M. Plan and resource-bounded practical reasoning. *Computational Intelligence 4* (1988), 349–355.

[2] García, A. J., and Simari, G. R. Defeasible logic programming: An argumentative approach. In *Theory and Practice of Logic Programming* (2004), pp. 95–138.

[3] Jennings, N. R., Parsons, S., Noriega, P., and Sierra, C. On argumentation-based negotiation. In *Proc. of the Int. Workshop on Multi-Agent Systems* (Boston, USA, 1998).

[4] Labrou, Y. *Semantics for an Agent Communication Language*. PhD thesis, Department of Computer Science and Electrical Engineering, Univ. of Maryland and Baltimore County, 1996.

[5] Müller, H. J. Negotiation principles. In *Foundations of Distributed Artificial Intelligence*, G. O'Hare and N. Jennings, Eds. John Wiley and Sons, 1996, ch. 7.

[6] Parsons, S., Sierra, C., and Jennings, N. Agents that reason and negotiate by arguing. *Journal of Logic and Computation 8*, 3 (June 1998), 261–292.

[7] Rueda, S. V., García, A. J., and Simari, G. R. Argument-based negotiation among bdi agents. *Journal of Computer Science and Technology 2*, 7 (2002), 1–8.

[8] Rueda, S. V., García, A. J., and Simari, G. R. Alternativas para la representación del conocimiento compartido entre agentes colaborativos. In *Proc. of the VIII Congreso Argentino de Ciencias de la Computación* (2004), pp. 1508–1519.

[9] Searle, J. R. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge, United Kingdom, 1969.

[10] Wooldridge, M. Intelligent agents. In *Multiagent Systems A Modern Approach to Distributed Artificial Intelligence*, G. Weiss, Ed. MIT Press, 2000, pp. 27–77.