Interaction among BDI Argumentative Agents: A Dialogue Games Approach

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

Negotiation is a fundamental activity in a multi-agent system. The members of the system negotiate in order to coordinate their activities and to distribute resources and tasks trying to reach a state acceptable to all.

Carrying out a negotiation process requires that agents be able to interact. In this work, protocol interaction is implemented by means of *dialogues* between agents, and the set of dialogues generated inside the same negotiation process conforms a *conversation*. We propose an interaction language that allows argumentation-based negotiation among collaborative BDI agents. For the language description we use a formalism, called *Dialogue Games*, which allows to specify the nature of the utterances.

1 Introduction

Negotiation is a fundamental activity in a multi-agent system. The members of the system negotiate in order to coordinate their activities and to distribute resources and tasks trying to reach a state acceptable to all. The negotiation models vary depending on the system's characteristics. If all the members are part of an organization, the relationship among them can be a collaborative one, even when it will be frequently necessary for them to interact in order to align their interests. The group can also be composed of homogeneous or heterogeneous agents. In the former case, all the members share the same view of the world and they have identical capacities. In a heterogeneous group, agents will in general have distinct views of the world and different abilities.

In this work we adopt the BDI model for representing the mental attitudes of each member of the group. The individual knowledge of each agent is conformed by its specific knowledge and the knowledge shared with other members in the group; each agent will reason using the facts that are available to it. As it is proposed in [10], the shared knowledge is distributed among *pairs* of agents; therefore, although each agent's view of the world is consistent, different

[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, PICT 2003 15043 and PAV 076)

[2] Member of the IICyTI (Instituto de Investigación Científica en Ciencia y Tecnología Informática).

members of the system can have different views. The group is heterogeneous, and each agent's goals are tied to their abilities. Despite their differences, all of the members in the organization are autonomous and rational entities with a collaborative attitude. When their beliefs and abilities do not suffice to reach their goals they request collaboration, starting a negotiation process.

Argument-based negotiation is a suitable alternative for modelling situations in which agents have limited information and bounded capacities [9]. During the process, the participants acquire information, but it is also possible for them to reach a point in which they must revise their plans and even modify their preferences in order to be able to reach an agreement. In our approach each agent elaborates arguments as part of its own planning processes [2] and to justify its proposals, counter-proposals, and rejections during the negotiation process. Interaction is implemented by means of *dialogues* between agents, and the set of dialogues generated inside the same negotiation process conforms a *conversation.* The structure of a dialogical system can be thought of as a dialogical game, in the sense that two participants perform movements by taking turns following their own goals and strategies.

In the next section we present an interaction model among collaborative agents based on dialogues and conversations. For the specification we use a formalism, called *Dialogue Games*, which allows to describe the nature of the utterances available in each dialogue. In section 3 we propose an interaction language which favours the argumentation-based negotiation among BDI agents. For each locution's specification we establish a set of preconditions, its meaning, the expected responses, and the modifications that it could produce. The interaction protocol and the interaction language as a whole can be used for specifying meaningful interaction between dialogical partners by following the rules of an individual dialogue. In the last section we discuss conclusions and outline future work.

2 Interaction Protocol

A multi-agent system consists of a group of agents that interact with each other. This interaction is generally regarded as the foundation for cooperative and competitive behaviour in autonomous agents. The term *interaction protocol* is used in reference to a set of rules that guide interactions. Throughout this work, we use a special kind of interaction called *negotiation*, which allows agents interacting with each other to reach a mutually acceptable agreement.

Negotiation can be thought of as a distributed search process over the space of potential agreements. The process is not linear, and therefore the space is not reduced until the solution is reached because it can move and even incorporate new points. In most cases, each agent knows only part of the search space and, within it, there is only a portion which satisfies its expectations. Each agent has a specific set of points within the space of agreements that are acceptable to it. The search is successful when an *agreement space* is reached, that is, there exists a nonempty intersection among the individual spaces. The process ends when the search ends, regardless of its success or failure.

In a simple interaction protocol the agents elaborate, accept, or reject proposals. This approach is not adequate when negotiation is viewed as a search process. In this case, the receiver of a proposal must be able not only to accept or reject the proposal, but also to guide the process with its answer. Agents perform proposals and counter-proposals elaborating *arguments* which intend to persuade other agents [4]. The interaction language must then offer a set of primitives suitable for expressing proposals and counter-proposals, offering arguments and expressing the interest level that agents assign to each collaboration request.

Our model proposes that agents' collaboration requests be restricted only to requests for other agents' beliefs, and the possibility of requesting the execution of an action is not considered (at least not directly through a primitive). When an agent requests collaboration for a certain literal p it is indicating that it needs to include p in its own knowledge, and another agent must add it on its behalf. When the negotiation ends successfully, the shared knowledge is modified with the incorporation of new beliefs. The shared knowledge is distributed among pairs of agents, and therefore the modification initially affects only two agents. However, the negotiation process may have involved various members of the group. Thus, the language must allow agents to manifest not only their expectation that another member of the group remain committed to the negotiation, but also their intention to free such agent from that commitment.

2.1 Dialogues and Conversations

Interaction in this model is implemented by means of dialogues between two agents. A *dialogue* is a sequence of speech acts exchanged between two agents that share the same aim and intend to follow a turn-taking scheme. Each participant may also have its own goals in the dialogue, which should balance with the commitment imposed by the shared goal. Each type of dialogue requires certain level of commitment and argumentation, and each participant has associated a set of propositions which composes its set of agreements. As the dialogue evolves, each interlocutor's set of agreements is modified in order to add or remove propositions. Starting from a single dialogue, the negotiation process derives 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 [11], where agents establish a dialogue with another agent and the negotiation process consists of a single dialogue.

If an agent, let's say $\mathbf{a}_1$, needs collaboration regarding some literal p, it performs a global collaboration request, to which agents in the system will respond showing their willingness to consider a specific request. When another agent, for example $\mathbf{a}_2$, receives a message from agent $\mathbf{a}_1$, requesting collaboration for a specific literal, a dialogue has begun between $\mathbf{a}_1$ and $\mathbf{a}_2$, by which they will negotiate the obtention of p. In order to collaborate, agent $\mathbf{a}_2$ 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 $\mathbf{a}_1$ cannot attend this counter-proposal, agent $\mathbf{a}_2$ can make the request to another agent in the group, as can be agent $\mathbf{a}_3$, suspending the dialogue with $\mathbf{a}_1$ and establishing a new one with $\mathbf{a}_3$.

The dialogue between 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 $\mathbf{a}_1$ and agent $\mathbf{a}_2$, 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 [10]. The existence of conflicts between agents $\mathbf{a}_2$ and $\mathbf{a}_3$ can also prevent the successful termination of the dialogue between $\mathbf{a}_1$ and $\mathbf{a}_2$. In these cases, our negotiation model proposes that agent $\mathbf{a}_2$ should inform agent $\mathbf{a}_1$ that the conflict with agent $\mathbf{a}_3$ is preventing it from collaborating; consequently, agent $\mathbf{a}_1$ must be the one which insists agent $\mathbf{a}_3$ 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 first two agents that appear are the sender and the receiver of the message, and the dialogue involves only them.

2.2 Dialogue Games

The literature offers different formalisms for specifying interaction protocols in multi-agent systems. No matter which alternative is chosen, it must at least include the following elements:

- Types of participants.
- Interaction states.
- Events which trigger state changes.
- Valid actions given the participant and the state.

Dialogue games are a particular alternative suitable for expressing argumentation. This formalism can be used to specify meaningful interaction between dialogical partners by following the rules of an individual dialogue. The interaction between two or more *players* is defined by means of a formal dialogue game, in which locutions are considered to be moves. The rules specify which locutions are permitted under what circumstances, and which responses are possible. There are different types of dialogue game rules, as proposed in [7]:

- Commencement and termination: define the circumstances under which the dialogue begins and ends.
- Locutions: specify the nature of the utterances permitted in the dialogue.
- Combination: define the dialogical contexts under which a particular locution is allowed.
- Commitment: define the circumstances under which a participant expresses dialogical commitment to a proposition.

This formalism provides a unifying framework that represents different types of dialogues, each of which has a simple semantics. In an interaction protocol based on dialogue games, it is possible to identify appropriate speech acts and to define constraints on their utterances. Basically, the goal of a dialogue model is to structure a dialogue into *dialogue acts*, and to find the relationships between dialogue acts -utterances- that explain its coherence. A dialogue is said to be *coherent* if the sequence of utterances performed by the participants builds a dialogue context which represents the set of statements and commitments that were made by them [3].

The role of the language is essential in the development of software applications based on the agents model. Therefore, the specification of a *communication language* is necessary. Two key elements are required for this purpose: an *interaction language* and an *interaction protocol.* In the next section we present an interaction language for a multi-agent system whose members are argumentative BDI agents.

3 Interaction Language

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 proposed in [5], is the communication of messages which represent the agent's knowledge and that are interpreted in a well defined manner. Furthermore, these messages cause certain actions on behalf of both the sender and the receiver.

An interaction language based on *Speech Act Theory* is composed of a set of locutions or utterances -communication primitives-. An utterance is a single meaningful unit of communication [3], and it is composed of a *semantic content*, i.e. the information conveyed in it, and a *communicative purpose.* An interaction language for argumentation-based negotiation among BDI agents is proposed in [10]. In the following we present an extension for this language and we show a dialogue game based specification for it.

3.1 Locution specification

In the following we will consider $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$ as agents in the system, p as a literal, and Q as a set of literals.

- Initiation of the Interaction Process
  - Locution: `Request_coll(`$\mathbf{a}_1$`)`.
    - **Preconditions:** Agent $\mathbf{a}_1$ must need a literal for which it cannot build a warranty from its individual knowledge, nor elaborate a plan that allows it to add the literal to its individual knowledge.
    - **Meaning:** Agent $\mathbf{a}_1$ asks the rest of the members in the system which of them are available for considering a collaboration request.
    - **Response:** Any agent in the system which is available for considering a collaboration request, may respond with an appropriately instantiated *Available*() locution.
    - **Updates:** No effects.
  - Locution: `Available(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`)`.

- **Preconditions:** Participant $\mathbf{a}_2$ must have previously uttered a locution *Request_coll* in the system.
- **Meaning:** Agent $\mathbf{a}_1$ lets agent $\mathbf{a}_2$ know that it is available for considering its collaboration request.
- **Response:** None required.
- **Updates:** Agent $\mathbf{a}_2$, the one which uttered the *Request_coll* locution, must store the names of all agents that respond to the global request with an *Available*$(\mathbf{a}_1, \mathbf{a}_2)$ locution, constituting the set of all the possible agents that can intervene in the initiated conversation.

- Collaboration Requests
  - Locution: $\texttt{Request(}\mathbf{a}_1\texttt{, }\mathbf{a}_2\texttt{, }\mathbf{a}_3\texttt{, }p\texttt{, }Q\texttt{,}\mathit{willingAgentSet}\texttt{)}$, where *willingAgentSet* is the set of all agents that are willing to cooperate in this conversation.
    - **Preconditions:** Agent $\mathbf{a}_1$ has a goal for which it has built a plan, but there exists a literal p necessary for this plan which $\mathbf{a}_1$ cannot obtain by itself. That is, it must be impossible for $\mathbf{a}_1$ to build a warranty for p, or elaborate a plan to obtain it. Before uttering this locution, agent $\mathbf{a}_1$ must have uttered a *Request_Coll* locution, and agent $\mathbf{a}_2$ must be a member of the *willingAgentSet*.
    - **Meaning:** If the third argument is not present the meaning of this locution is that agent $\mathbf{a}_1$ needs p and requests $\mathbf{a}_2$'s collaboration in order to obtain it. In the other case -the third argument is instantiated-, agent $\mathbf{a}_1$ needs p and requests $\mathbf{a}_2$'s collaboration in order to obtain it, but this time the request is on behalf of agent $\mathbf{a}_3$. In both cases argument Q conforms the set of beliefs that must be avoided in the plan for obtaining p.
    - **Response:** Agent $\mathbf{a}_2$ will try to build a plan for p. It will respond:
      - Accept(), if it finds a plan for p and it has all that it needs in order to effectively obtain p;
      - Unable(), if it cannot find a plan for p, that is its knowledge and capabilities do no suffice;
      - Reject(), if it finds a plan for p but some of the preconditions needed -or even p- are in conflict with its own goals or beliefs;
      - Indirect_Reject(), if it finds a plan for p but it is not able to obtain some of the literals needed in that plan by itself. Agent $\mathbf{a}_2$ asks another agent for help, but this agent cannot do so because it has conflicts.
      - Request(), if it finds a plan for p but, in order to effectively obtain it, it needs another literal which cannot be obtain by itself. The agent must ask $\mathbf{a}_1$ for help but, if $\mathbf{a}_1$ is not able to help it or if $\mathbf{a}_1$ does not respond, it should choice one *willingAgentSet*'s member in order to continue the negotiation.
    - **Updates:** If an agent utters this locution as a counter-proposal for literal p, it must store the plan founded for p, along with the plan's requirements -preconditions-, while it waits for help.

- Locution: `Insist(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`, `$\mathbf{a}_3$`, `p`, `*willingAgentSet*`)`, where *willingAgentSet* is the set of all agents that are willing to cooperate in this conversation.
  - **Preconditions:** Agent $\mathbf{a}_2$ must have previously uttered a *Reject* or *Indirect_Reject* locution to agent $\mathbf{a}_1$ referring to a *Request* locution for literal p. In this case, agent $\mathbf{a}_1$ should revise its initial plan for which it needed p. If $\mathbf{a}_1$ cannot find a way to avoid needing p, it should request p again but this time with more vehemence.
  - **Meaning:** If the third argument is not present the meaning of this locution is that agent $\mathbf{a}_1$ asks agent $\mathbf{a}_2$ to revise its plans in order to avoid any conflict with p. In the other case -the third argument is instantiated-, agent $\mathbf{a}_1$ needs p and insists on $\mathbf{a}_2$'s collaboration in order to obtain it, but this time the request is on behalf of agent $\mathbf{a}_3$.
  - **Response:** Agent $\mathbf{a}_2$ will revise its plans in order to obtain p. It will respond:
    - Accept(), if it finds another plan for p and it has all that it needs in order to effectively obtain it;
    - Reject(), if it cannot find another plan for p or it find one but some of the preconditions needed -or even p- are in conflict with its own goals or beliefs;
    - Indirect_Reject(), if it finds another plan for p but it is not able to obtain some of the literals needed in that plan by itself. Agent $\mathbf{a}_2$ asks another agent for help, but this agent cannot do so because it has conflicts.
    - Request(), if it finds a plan for p but, in order to effectively obtain it, it needs another literal which cannot be obtain by itself. The agent must ask $\mathbf{a}_1$ for help but, if $\mathbf{a}_1$ is not able to help it, it should choice one *willingAgentSet*'s member in order to continue the negotiation.
  - **Updates:** No effects.
- Locution: `Demand(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`, `$\mathbf{a}_3$`, `p`, `*willingAgentSet*`)`, where *willingAgentSet* is the set of all agents that are willing to cooperate in this conversation.
  - **Preconditions:** Agent $\mathbf{a}_2$ must have previously uttered a *Reject* or *Indirect_Reject* locution to agent $\mathbf{a}_1$ referring to an *Insist* locution for literal p. In this case, agent $\mathbf{a}_1$ should revise its goals. If it still needs p after the revision, it should demand collaboration for obtaining p.
  - **Meaning:** If the third argument is not present, the meaning of this locution is that agent $\mathbf{a}_1$ asks agent $\mathbf{a}_2$ to revise its goals in order to avoid any conflict with p. In the other case -the third argument is instantiated-, agent $\mathbf{a}_1$ needs p and demands $\mathbf{a}_2$'s collaboration in order to obtain it but this time the request is on behalf of agent $\mathbf{a}_3$.
  - **Response:** Agent $\mathbf{a}_2$ will revise its goals so it can obtain p. It will respond:
    - Accept(), if its goals are no longer in conflict with p and it has all that it needs in order to effectively obtain p;
    - Reject(), if after having revised its goals, it has conflicts in obtaining p;

      - Indirect_Reject(), if after having revised its goals, it needs help from another agent but this agent cannot do so because it has conflicts.
      - Request(), if it finds a plan for p but, in order to effectively obtain it, it needs another literal which cannot be obtain by itself. The agent must ask $\mathbf{a}_1$ for help but, if $\mathbf{a}_1$ is not able to help it, it should choice one *willingAgentSet*'s member in order to continue the negotiation.
    - **Updates:** No effects.
  - Locution: $\texttt{Still\_Int(}\mathbf{a}_1\texttt{, }\mathbf{a}_2\texttt{, }p\texttt{).}$
    - **Preconditions:** There must exists an open dialogue between agents $\mathbf{a}_1$ and $\mathbf{a}_2$. In this dialogue, agent $\mathbf{a}_1$ must have requested collaboration, regarding p, to agent $\mathbf{a}_2$, but it is not possible for agent $\mathbf{a}_2$ to accept it because it needs some elements that are not available to it and another agent is preventing its obtention. That is, agent $\mathbf{a}_2$ must have uttered an *Indirect_Reject* with respect a request for p performed by $\mathbf{a}_1$.
    - **Meaning:** 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. Agent $\mathbf{a}_1$ uttered this locution because it will suspend the dialogue with $\mathbf{a}_2$ in order to begin a new dialogue with another agent which is preventing them to reach an agreement.
    - **Response:** None required.
    - **Updates:** The internal state of both $\mathbf{a}_1$ and $\mathbf{a}_2$ changes. Agent $\mathbf{a}_2$ becomes committed to being available for obtaining p, that is, agent $\mathbf{a}_2$ is not allowed to change anything in its individual knowledge that could prevent it from obtaining p in the future. Agent $\mathbf{a}_1$ has the responsibility of informing whether it will actually make the request or free $\mathbf{a}_2$ from the commitment.
  - Locution: $\texttt{Free(}\mathbf{a}_1\texttt{, }\mathbf{a}_2\texttt{, }p\texttt{).}$
    - **Preconditions:** Agent $\mathbf{a}_1$ must have previously uttered a *Still_int* locution committing agent $\mathbf{a}_2$ to the request for literal p.
    - **Meaning:** Agent $\mathbf{a}_1$ indicates to $\mathbf{a}_2$ that it is free from the request done for p.
    - **Response:** None required.
    - **Updates:** Agent $\mathbf{a}_2$ is no longer engaged with agent $\mathbf{a}_1$ regarding the collaboration in obtaining literal p.

- Responses to a collaboration request
  - Locution: $\texttt{Accept(}\mathbf{a}_1\texttt{, }\mathbf{a}_2\texttt{, }p\texttt{).}$
    - **Preconditions:** Agent $\mathbf{a}_2$ must have previously uttered a *Request*$(\mathbf{a}_2, \mathbf{a}_1, p)$, *Insist*$(\mathbf{a}_2, \mathbf{a}_1, p)$, or *Demand*$(\mathbf{a}_2, \mathbf{a}_1, p)$ locution. Agent $\mathbf{a}_1$ must have p in its individual knowledge, have a warranty for it, or be able to build a plan for

obtaining it. This locution cannot be uttered after an *Unable*$(\mathbf{a}_1, \mathbf{a}_2, p)$ or after a *Reject*$(\mathbf{a}_1, \mathbf{a}_2, p)$ locution. It is necessary for agent $\mathbf{a}_1$ to utter a *Done*$(\mathbf{a}_1, \mathbf{a}_2, p)$ locution after having uttered this locution. At this moment, the interaction is not a turn-taking dialogue, agent $\mathbf{a}_1$ will utter two locutions in a row.

  - ∗ **Meaning:** 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. Furthermore, there are no conflicts in adding p to the shared knowledge.
  - ∗ **Response:** None required.
  - ∗ **Updates:** Agent $\mathbf{a}_1$ is committed to eventually obtain p to satisfy $\mathbf{a}_2$'s request.

- Locution: `Unable(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`, `p`)`.
  - ∗ **Preconditions:** Agent $\mathbf{a}_2$ must have previously uttered a *Request*$(\mathbf{a}_2, \mathbf{a}_1, p)$ locution. Agent $\mathbf{a}_1$ must not be able to build a plan or obtain a warranty for p. This locution cannot be uttered after an *Accept*$(\mathbf{a}_1, \mathbf{a}_2, p)$ or after a *Reject*$(\mathbf{a}_1, \mathbf{a}_2, p)$ locution.
  - ∗ **Meaning:** Agent $\mathbf{a}_1$ informs agent $\mathbf{a}_2$ that it is not capable of obtaining p.
  - ∗ **Response:** None required.
  - ∗ **Updates:** With this utterance, the dialogue between $\mathbf{a}_1$ and $\mathbf{a}_2$ has finished unsuccessfully. After uttering this locution, $\mathbf{a}_2$ should not utter an *Insist*$(\mathbf{a}_2, \mathbf{a}_1, p)$ or *Demand*$(\mathbf{a}_2, \mathbf{a}_1, p)$ locution because $\mathbf{a}_1$ will never be capable of accepting its request.

- Locution: `Reject(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`, `p`)`.
  - ∗ **Preconditions:** Agent $\mathbf{a}_2$ must have previously uttered a *Request*$(\mathbf{a}_2, \mathbf{a}_1, p)$, *Insist*$(\mathbf{a}_2, \mathbf{a}_1, p)$, or *Demand*$(\mathbf{a}_2, \mathbf{a}_1, p)$ locution. There must exist conflicts between $\mathbf{a}_1$'s plans or goals and $\mathbf{a}_2$'s request about literal p. This locution cannot be uttered after an *Unable*$(\mathbf{a}_1, \mathbf{a}_2, p)$ locution.
  - ∗ **Meaning:** Agent $\mathbf{a}_1$ informs agent $\mathbf{a}_2$ that there exists a conflict between its own plans and p.
  - ∗ **Response:** If the locution corresponds to a *Request*$(\mathbf{a}_2, \mathbf{a}_1, p)$ locution, agent $\mathbf{a}_2$ may respond with *Insist*$(\mathbf{a}_2, \mathbf{a}_1, p)$. If the locution correspond to an *Insist*$(\mathbf{a}_2, \mathbf{a}_1, p)$ locution, agent $\mathbf{a}_2$ may respond with *Demand*$(\mathbf{a}_2, \mathbf{a}_1, p)$.
  - ∗ **Updates:** Agent $\mathbf{a}_1$ has rejected $\mathbf{a}_2$'s proposal because there exist conflicts between the plan it found for p and its own goals or beliefs. However, it must store this plan because it is possible for $\mathbf{a}_2$ to insist on p; in this case, $\mathbf{a}_1$ will need the plan in order to revise it against its goals and beliefs.

- Locution: `Indirect_Reject(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`, `$\mathbf{a}_3$`, `p`, `Q`)`.
  - ∗ **Preconditions:** Agent $\mathbf{a}_2$ must have previously uttered a *Request*$(\mathbf{a}_2, \mathbf{a}_1, p)$, *Insist*$(\mathbf{a}_2, \mathbf{a}_1, p)$, or *Demand*$(\mathbf{a}_2, \mathbf{a}_1, p)$ locution. There must exist literals in Q that $\mathbf{a}_1$ needs in order to build a plan or a warranty for p but agent $\mathbf{a}_3$ has conflicts their addition to the shared knowledge or with helping agent $\mathbf{a}_1$ in obtaining them. This locution cannot be uttered after an *Unable*$(\mathbf{a}_1, \mathbf{a}_2, p)$ locution.

    - **Meaning:** 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.
    - **Response:** If agent $\mathbf{a}_2$ decides to continue the negotiation with agent $\mathbf{a}_3$ by itself then it may respond with a $\mathit{Still\_Int}(\mathbf{a}_2, \mathbf{a}_1, p)$ locution in order to commit agent $\mathbf{a}_1$'s availability for accepting the request.
    - **Updates:** Agent $\mathbf{a}_1$ has rejected $\mathbf{a}_2$'s proposal because there exist conflicts between the plan it found for p and the goals or beliefs of another agent. However, it must store this plan because it is possible for $\mathbf{a}_2$ to insist on p; in this case, $\mathbf{a}_1$ will need the plan in order to revise it against its goals and beliefs.
  - Locution: `Done(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`, `p`).`
    - **Preconditions:** Agent $\mathbf{a}_1$ must have uttered an $\mathit{Accept}(\mathbf{a}_1, \mathbf{a}_2, p)$ locution.
    - **Meaning:** Agent $\mathbf{a}_1$ informs agent $\mathbf{a}_2$ that it has performed all the necessary actions in order to obtain p and that the literal has been added to the shared knowledge.
    - **Response:** None required.
    - **Updates:** Shared knowledge between agent $\mathbf{a}_1$ and $\mathbf{a}_2$ is updated with the addition of literal p. After uttering this locution, the dialogue between these agents has finished successfully.

- Requests for the modification of shared knowledge
  - Locution: `Request_add(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`, `p`).`
    - **Preconditions:** Agent $\mathbf{a}_1$ has elaborated a plan for the literal p or for some other literal that it needs, not only for obtaining p but also for reaching other individual goals. Whatever was the case, the execution of the plan requires the addition of the literal to the shared knowledge between agents $\mathbf{a}_1$ and $\mathbf{a}_2$.
    - **Meaning:** Agent $\mathbf{a}_1$ asks agent $\mathbf{a}_2$ for authorization to add p to their shared knowledge.
    - **Response:** Authorize_add($\mathbf{a}_1$, $\mathbf{a}_2$, p) or Reject_add($\mathbf{a}_1$, $\mathbf{a}_2$, p).
    - **Updates:** No effects.
  - Locution: `Authorize_add(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`, `p`).`
    - **Preconditions:** Agent $\mathbf{a}_2$ must have uttered a $\mathit{Request\_add}(\mathbf{a}_2, \mathbf{a}_1, p)$ locution. There must not exist conflicts between $\mathbf{a}_1$'s goals or plans and the addition of literal p to the knowledge shared with agent $\mathbf{a}_2$.
    - **Meaning:** 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.
    - **Response:** None required.

    - **Updates:** After uttering this locution, agent $\mathbf{a}_1$ can add literal p to the knowledge shared with agent $\mathbf{a}_2$.

- Locution: `Reject_add(`$\mathbf{a}_1$`, `$\mathbf{a}_2$`, `p`).`
    - **Preconditions:** Agent $\mathbf{a}_2$ must have uttered a $\textit{Request\_add}(\mathbf{a}_2, \mathbf{a}_1, p)$. There must exist conflicts between $\mathbf{a}_1$'s goals or plans and the addition of literal p to the knowledge shared with agent $\mathbf{a}_2$.
    - **Meaning:** 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.
    - **Response:** None required.
    - **Updates:** No effects.

3.2 Commencement and Termination Rules

A conversation among agents in the system starts when one of them performs a global collaboration request, and ends when all the dialogues that it derived finish. A conversation is said to have finished successfully if the particular performed request is satisfied; this does not guarantee that all the dialogues derived in the conversation have ended successfully.

A dialogue between two agents begins when the one which needs help for obtaining some literal necessary as a precondition for a plan that it has built performs a specific request to a particular agent. Both agents enter a negotiation process in order to obtain the requested literal and, as we explained above, other agents can also intervene in the process generating new dialogues in the conversation. The dialogue can finish successfully or not; it finishes successfully if the agents reach an agreement and the requested literal is added to the knowledge shared between them. On the other hand, it is possible for them not to reach an agreement, a situation which can arise due either to conflicts with respect to goals or plans, or to lack of information or capabilities of the agent which is trying to help. If this situation arises or the conflicts are not resolved, the dialogue ends unsuccessfully.

4 Conclusions and Future Work

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

In this work we proposed an interaction language based on Speech Act theory, where the basic characteristics of human communication are captured and represented in a model suitable for artificial agents. We also present an interaction protocol which gives a set of rules for determining the structures of the dialogues generated during the negotiation process.

Our future work is oriented towards the analysis of alternatives in the specification of conversations. Graphic specification languages are particularly attractive because they allow making the connection among dialogues explicit. UML is currently one of the most powerful graphic design languages for describing software systems, and it provides activity diagrams that can be used for specifying the interaction among the agents in a system. Computations are expressed in terms of states and the progression through them, and the main components used in the description of interaction protocols are action states, activity states, and transitions.

A large body of research proposes an extension of UML, increasing its expressive power in order to support concepts which are specifically oriented towards interaction among agents; other authors consider that it is important to maintain only one general graphic language. Our proposal is to model conversations by means of activity diagrams in UML in order to reflect the structure of the set of dialogues and their interactions.

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] Hulstijn, J., Torre, L., and Dastani, M. Negotiation protocols and dialogue games, 2000.

[4] 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).

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

[6] Levin, J. A., and Moore, J. A. Dialogue-games: Metacommuncation structures for natural language interaction. *Cognitive Science 1*, 4 (Oct. 1977), 395–420. (Also in *Readings in Distributed Artificial Intelligence*, Alan H. Bond and Les Gasser, eds, pp. 385–397, Morgan Kaufmann, 1988.).

[7] Mcburney, P., Eijk, R. M. V., Parsons, S., and Amgoud, L. A dialogue game protocol for agent purchase negotiations. *Autonomous Agents and Multi-Agent Systems 7*, 3 (2003), 235–273.

[8] 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.

[9] Rahwan, I. Argumentation-based negotiation, 1985.

[10] Rueda, S., García, A., and Simari, G. Alternativas para la representación del conocimiento compartido entre agentes colaborativos. In *Proc. of the VIII CACIC* (2004), pp. 1508–1519.

[11] 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.

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

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