

An Argumentative Model for Service-Oriented Agents

Maxime Morge*, Paolo Mancarella*
Jarred McGinnis**, Stefano Bromuri** and Kostas Stathis**

*Università di Pisa

via F. Buonarroti, 2 I-56127 Pisa, Italy
{morge,paolo}@di.unipi.it

**Royal Holloway College, University of London
Egham, Surrey TW20 0EX, UK
{jarred,stefan,kostas}@cs.rhul.ac.uk

Abstract

In this paper we propose an argumentative agent model able to reason and make decisions, communicate and negotiate with other agents with the aim of supporting service selection and composition. Actually, our agent model is inspired from the general-purpose KGP agent model which deals only partially with priorities as required by our application. We provide here an argumentation framework which adopts Knowledge, Goals, Decisions, and Priorities as the main component to perform the individual and social reasoning of agents. We propose here an example to illustrate our approach.

Introduction

Service-oriented programming is proving to be an excellent test bed by requiring agents to play various roles for the provision of services in an open and distributed system (Hendler 2001). On the other side, argumentation provides a powerful agent model for interacting, deliberating agents to assess the validity of received information and to be able to resolve conflicts of opinion. It is an essential ingredient of decision-making (Kakas & Moraitis 2003; Morge & Mancarella 2007), inter-agent communication, and negotiation (Kakas & Moraitis 2006; Amgoud, Dimopoulos, & Moraitis 2007).

In this paper we propose an argumentative agent model able to reason, communicate and negotiate with other agents with the aim of supporting service selection/composition, as envisaged in the ARGUGRID project¹. The human user requesting (resp. controlling) a service is only required to specify an abstract description of her needs (resp. competencies) for this service, possibly with some preferences about them. The selection (resp. the suggestion) of these services, as well as their composition, are tasks delegated to the autonomous agents. The main focus and contribution of this paper is to provide an in-depth report on our agent model inspired from the general-purpose KGP model (Kakas *et al.* 2004). This latter deals only partially with priorities as required by our application. We provide here a suitable representation of Knowledge, Goals, Decisions

and Priorities built upon the approach of (Dung, Mancarella, & Toni 2007), which does not deal only with the preferences amongst goals but also with uncertainty of knowledge and expected utilities of decisions. In this way, we provide a computational argumentation framework (AF, for short) which performs the individual reasoning, allowing to shift from the goals and preferences provided by the user requesting (resp. providing) a service to an internal and abstract representation of her needs (resp. competencies). Moreover, this AF performs the social reasoning, allowing to shift, by means of negotiation, from these abstract representations to concrete ones, in terms of contracts. A protocol engine manages the communication. Finally, we deploy a typical procurement process to illustrate our approach.

The paper is organized as follows. At first, we introduce the real world use case to motivate/illustrate our approach. Secondly, we present our computational argumentation framework for decision making. Thirdly, we apply our AF to perform the individual reasoning and the social reasoning of agents. Then, we outline the social interaction amongst agents. Finally, we discuss related work and we conclude by summarising our proposal.

Use Case Scenario

In order to illustrate our model, we consider an e-procurement scenario where a buyer seeks to purchase complex services and the suppliers combine their competencies in order to provide solutions. We consider 6-steps procurement processes (Stournaras 2007) whereby: a requester looks for potential suppliers (step 1), gathers information in order to evaluate them (step 2), creates a shortlist according to this information (step 3). The requester asks the short-listed suppliers to provide a quote for the services (step 4), chooses one of the suppliers (step 5), and finally negotiates with the winner the terms & conditions of the contract (step 6). Information-seeking dialogues are required for supporting the steps 1, 2 and 4 while argumentation-based negotiation is required to perform the step 6. These dialogues should conform to suitable protocols.

Within the generic e-procurement scenario we consider a specific case where a buyer looks for a service S which is composed of a service S_a , provided by an A-type agent, and a service S_b , provided by a B-type agent. We will focus on the process involving an A-type agent A_1 as a requester and

a B-type agent Bob as a supplier. A1's goal consists of finding and agreeing to a service S_b provided by a B-type agent. According to A1 preferences, the cost of the service must be low, and the quality of the service must be high. A1 needs to solve a decision-making problem about the service and the supplier for that service. On the other hand, the goal of the supplier Bob consists of agreeing to provide a service. According to Bob's preferences, the cost of the service must be high, and the quality of the service must be low. Bob needs to solve a decision-making problem about the service it can provide. The two decision making processes take place in a dynamic setting, whereby information about other agents and the services they require/provide is obtained incrementally within the e-procurement process described previously. The outcome of this process will be a contract, i.e. a legal relation between them that typically force commitments (e.g. obligations) from one agent to another about the provision of the services. In the concrete use case, the contract obliges Bob to provide the service $S_b(e)$ to A1, with low cost and low quality.

Within our proposed agent model, the e-procurement process is supported by argumentation. In the concrete use case, A1, as a requester, uses argumentation to collect information on the available services and on the suppliers. A1 (resp. Bob), as a requester, (resp. as a provider) uses argumentation for deciding which service it needs (resp. it can provide) taking into account its preferences and possibly the inconsistency of information it has gathered. Moreover, through argumentation, the participants provide an interactive and intelligible explanation of their choices. For instance, A1 can argue that a service is justified as a good deal from his viewpoint since its cost is low. The previous argument will incite Bob to suggest a service with a low cost to reach quickly a good deal. Thus, in our framework agents can use argumentation for influencing each other.

Argumentation Framework

We present here our computational argumentation framework for decision-making (Morge & Mancarella 2007). At first, we introduce argumentation. At second, we define the decision framework which captures decision making problems. At third, we define the arguments and their interaction.

Abstract Argumentation

The framework proposed in this paper is based on Dung's abstract approach to defeasible argumentation (Dung 1995) which considers arguments as atomic and abstract entities interacting through a single binary relation over these. More formally, an abstract argumentation framework is a pair $AAF = \langle \mathcal{A}, \text{defeats} \rangle$ where \mathcal{A} is a finite set of arguments and $\text{defeats} \subseteq \mathcal{A} \times \mathcal{A}$ is a binary relation² over \mathcal{A} . We say that a set S of arguments defeats an argument a if a is defeated by one argument in S .

According to this framework, Dung introduces various extension-based semantics in order to analyse whenever a set of arguments can be considered as collectively justified.

Definition 1 (Semantics) Let $\langle \mathcal{A}, \text{defeats} \rangle$ be an AAF. For $S \subseteq \mathcal{A}$ a set of arguments, we say that:

- S is conflict-free iff $\forall a, b \in S$ a does not defeat b ;
- S is admissible iff S is conflict-free and S defeats every argument a such that a defeats S ;
- S is preferred iff S is maximally admissible;
- S is complete iff S is admissible and S contains all arguments a such that S defeats all defeaters against a ;
- S is grounded iff S is minimally complete;

These declarative model-theoretic semantics of the AAF capture various degrees of justification ranging from very permissive conditions, called *credulous*, to restrictive requirements, called *sceptical*. The semantics of an admissible (or preferred) set of arguments is credulous. However, there might be several conflicting admissible sets. That is the reason why various sceptical semantics have been proposed for the AAF, notably the grounded semantics. Since some ultimate choices amongst various justified sets of alternatives are not always possible, we consider in this paper only the credulous semantics. Statements representing the decision making problem should be expressed in order to perform decision making.

Decision Framework

Since we want to instantiate our AF for performing the individual/social reasoning of agents, we need to specify a particular logic allowing to express statements representing knowledge, goals, and decisions as well as priorities amongst these statements.

In the object language \mathcal{L} , we distinguish:

- a set of *abstract goals* (resp. *concrete goals*), i.e. some predicate symbols which represent the abstract (resp. concrete) features of the decisions;
- a set of *decisions*, i.e. some predicate symbols which represent the actions which must be performed or not;
- a set of *alternatives*, i.e. some constants symbols which represent the mutually exclusive actions for each decision;
- a set of *beliefs*, i.e. some predicate symbols which represent epistemic statements. In the language, we explicitly distinguish *assumable* (resp. *non-assumable*) beliefs, i.e. literals which can (resp. cannot) be taken for granted.

We assume that users provide, via the GUI, *influence diagrams* (cf. examples in Fig. 1), i.e. simple graphical representations of multi-attribute decision problems, to display the structure of the problem related to the evaluation of services (to be obtained or provided). In addition, the GUI allows the user to communicate user-specific preferences. The elements of the decision problem, i.e. *values* (represented by rectangles with rounded corners), *decisions* (represented by rectangles) and *knowledge* (represented by ovals), are connected by arcs where predecessors are independent and affect successors. Values may be structured hierarchically in an influence diagrams, where the top, *abstract* values (represented by rectangles with rounded corner and double line) aggregate the lower, *concrete* values. Values represent the agents' goals.

²Actually, the defeat relation is called attack in (Dung 1995).

Since we want to consider conflicts in the object language, we need some forms of negation. For this purpose, we consider strong negation, also called classical negation, and weak negation, also called negation as failure. A strong literal is an atomic first-order formula, possibly preceded by strong negation \neg . A weak literal is a literal of the form $\sim L$, where L is a strong literal. In order to express the mutual exclusion between statements, we define the incompatibility relation (denoted by \mathcal{I}) as a binary relation over atomic formulas which is asymmetric. Whatever the atom L is a belief or a goal, we have $L \mathcal{I} \neg L$ and $\neg L \mathcal{I} L$, while we have $L \mathcal{I} \sim L$ but we do not have $\sim L \mathcal{I} L$. Obviously, $D_1(a_1) \mathcal{I} D_1(a_2)$ and $D_1(a_2) \mathcal{I} D_1(a_1)$, D_1 being a decision predicate, a_1 and a_2 being different alternatives for D_1 . Notice that in general a decision can be addressed by more than two alternatives. We say that two sets of sentences Φ_1 and Φ_2 are incompatible ($\Phi_1 \mathcal{I} \Phi_2$) iff there is a sentence ϕ_1 in Φ_1 and a sentence ϕ_2 in Φ_2 such as $\phi_1 \mathcal{I} \phi_2$. A theory gathers the statements about the decision making problem.

Definition 2 (Theory) A theory \mathcal{T} is an extended logic program, i.e. a finite set of rules such as $R: L_0 \leftarrow L_1, \dots, L_j, \sim L_{j+1}, \dots, \sim L_n$ with $n \geq 0$, each L_i being a strong literal in \mathcal{L} . The literal L_0 , called the head of the rule, is denoted $head(R)$. The finite set $\{L_1, \dots, \sim L_n\}$, called the body of the rule, is denoted $body(R)$. The body of a rule can be empty. In this case, the rule is called a fact. R , called the name of the rule, is an atomic formula of \mathcal{L} .

Considering a decision making problem, we distinguish:

- *goal rules* of the form $R: G_0 \leftarrow G_1, \dots, G_n$ with $n > 0$. Each G_i is a goal literal in \mathcal{L} . The head of the rule is an abstract goal (or its strong negation). According to this rule, the abstract goal is promoted (or demoted) by the combination of goal literals in the body;
- *epistemic rules* of the form $R: B_0 \leftarrow B_1, \dots, B_n$ with $n \geq 0$. Each B_i is a belief literal of \mathcal{L} . According to this rule, the belief B_0 is true if the conditions B_1, \dots, B_n are satisfied;
- *decision rules* of the form $R: G \leftarrow D(a), B_1, \dots, B_n$ with $n \geq 0$. The head of the rule is a concrete goal (or its strong negation). The body includes a decision literal ($D(a) \in \mathcal{L}$) and a possible empty set of belief literals. According to this rule, the concrete goal is promoted (or demoted) by the decision $D(a)$, provided that conditions B_1, \dots, B_n are satisfied.

In order to evaluate the previous statements, all relevant pieces of information should be taken into account, such as the uncertainty of knowledge, the priority between goals, or the expected utilities of the decisions. For this purpose, we consider that the *priority* \mathcal{P} which is a (partial or total) pre-order on the rules in \mathcal{T} , i.e. a reflexive and transitive relation considering possible *ex aequo*. $R_1 \mathcal{P} R_2$ can be read “ R_1 has priority over R_2 ”. $R_1 \not\mathcal{P} R_2$ can be read “ R_1 has no priority over R_2 ”, either because R_1 and R_2 are *ex aequo* or because R_1 and R_2 are not comparable.

The priority over concurrent rules depends on the nature of rules. Rules are *concurrent* if their heads are identical or

incompatible. We define three priority relations over concurrent rules:

- the priority over *goal rules* comes from the *preferences* over goals. The priority of such rules corresponds to the relative importance of the combination of (sub)goals in the body as far as reaching the goal in the head is concerned;
- the priority over *epistemic rules* comes from the *uncertainty* of knowledge. The prior the rule is, the more likely the rule holds;
- the priority over *decision rules* comes from the *expected utility* of decisions. The priority of such rules corresponds to the expectation of the conditional decision in promoting/demoting the goal literal.

To summarize, a *decision framework* is a tuple $\mathcal{D} = \langle \mathcal{L}, \mathcal{I}, \mathcal{T}, \mathcal{P} \rangle$ which allows to capture a complex (and incomplete) representation about a decision making problem.

Structure of Arguments

Since we want that agents provide an intelligible explanation we adopt here the tree-like structure for arguments proposed in (Vreeswijk 1997) and we extend it with suppositions to build arguments upon missing information.

Definition 3 (Argument) An argument is composed by a conclusion, a top rule, some premises, some suppositions, and some sentences. These elements are abbreviated by the corresponding prefixes. An argument A is:

1. a hypothetical argument. If L is an assumable belief literal, then the argument built upon a ground instance of this assumable literal is defined as follows³:
 $conc(A) = L$, $top(A) = \theta$, $premise(A) = \emptyset$,
 $supp(A) = \{L\}$, $sent(A) = \{L\}$.
or
2. a built argument.
 - 2.1) If f is a fact in \mathcal{T} (i.e. $body(f) = \emptyset$), then the trivial argument A built upon this fact is defined as follows:
 $conc(A) = head(f)$, $top(A) = f$,
 $premise(A) = \emptyset$, $supp(A) = \emptyset$,
 $sent(A) = \{head(f)\}$.
 - 2.2) If r is a rule in \mathcal{T} , we define the tree argument A built upon this rule as follows. Let $body(r) = \{L_1, \dots, L_j, \sim L_{j+1}, \dots, \sim L_n\}$ and $sbarg(A) = \{A_1, \dots, A_n\}$ be the collection of arguments such that, for each strong literal $L_i \in body(r)$, $conc(A_i) = L_i$ with $i \leq j$ or $conc(A_i) = \sim L_i$ with $i > j$ (each A_i is called a subargument of A). Then:
 $conc(A) = head(r)$, $top(A) = r$,
 $premise(A) = body(r)$,
 $supp(A) = \cup_{A' \in sbarg(A)} supp(A')$,
 $sent(A) = \cup_{A' \in sbarg(A)} sent(A') \cup \{head(r)\} \cup body(r)$.

The subarguments of a tree argument concluding the weak literals in the body of the top rule are hypothetical arguments. Indeed, the conclusion of an hypothetical argument

³ θ denotes that no literal is required.

could be a strong or a weak literal while the conclusion of a built argument is a strong literal. As in (Vreeswijk 1997), we consider composite arguments, called *tree* arguments, and atomic arguments, called *trivial* arguments. Our definition considers that the different premises can be challenged and can be supported by subarguments. In this way, arguments are intelligible explanations. Moreover, we consider *hypothetical* arguments which are built upon missing information. In this way, our framework allows to reason further by making suppositions related to the unknown knowledge and over possible decisions.

Interaction

Since the sentences of arguments are conflicting, we define the attack relation amongst arguments.

Definition 4 (Attack relation) *Let A and B be two arguments. A attacks B iff $\text{sent}(A) \mathcal{I} \text{sent}(B)$.*

According to this definition, if an argument attacks a subargument, the whole argument is attacked.

Since arguments are more or less hypothetical, we define the size of their suppositions.

Definition 5 (Supposition size) *Let A be an arguments. The size of suppositions for A , denoted $\text{suppsz}(A)$, is the number of decision literals and assumable belief literals in the sentences of A .*

Since arguments have different natures (hypothetical or built) and the top rules of built arguments are more or less strong, we define the strength relation as follows.

Definition 6 (Strength relation) *Let A_1 be a hypothetical argument, and A_2, A_3 be two built arguments.*

1. A_2 is stronger than A_1 (denoted $A_2 \mathcal{P}^A A_1$);
2. If $(\text{top}(A_2) \mathcal{P} \text{top}(A_3)) \wedge \neg(\text{top}(A_3) \mathcal{P} \text{top}(A_2))$, then $A_2 \mathcal{P}^A A_3$;
3. If $(\text{top}(A_2) \mathcal{P} \text{top}(A_3)) \wedge (\text{suppsz}(A_2) < \text{suppsz}(A_3))$, then $A_2 \mathcal{P}^A A_3$;

Built arguments are preferred to hypothetical arguments. An argument is stronger than another argument if the top rule of the first argument has a proper higher priority than the top rule of the second argument, or if the top rules are incomparable but the number of suppositions made in the first argument is properly smaller than the number of suppositions made in the second argument.

The two previous relations can be combined.

Definition 7 (Defeats) *Let A and B be two arguments. A defeats B iff: i) A attacks B ; ii) $\neg(B \mathcal{P}^A A)$.*

In this section, we have defined the defeat relation in order to use the Dung's seminal calculus of opposition.

This argumentation framework, which has been already proposed in (Morge & Mancarella 2007), is implemented by MARGO⁴ by using the implementation of (Dung, Mancarella, & Toni 2007) in the CaSAPI system (Gartner & Toni 2007).

⁴<http://margo.sourceforge.net>

Individual Reasoning

We consider our AF to perform the individual reasoning which is about the kind of services which can be provided or requested. Decisions are made according to the individual statements, i.e. the user's requirements or competencies about the services, and the users' preferences.

In the influence diagram representing the competencies/needs of Al (cf top of Fig. 1) the main goal (provision) is split into independent, still abstract sub-goals: concerning the cost (cost) and the quality of the service (qos). These sub-goals are reduced to further, concrete sub-goals. The main goal, i.e. the provision of a composite service S , needs to be addressed by some decisions, e.g. on which concrete S_a and S_b service to adopt (by appropriately instantiating variables x and y in Fig. 1). Through the paper, variables are in italics and constants are in typescript font. Al may be able to provide several instances of S_a (x may be instantiated in different ways) and needs to choose one instance of service S_b (y may be instantiated in different ways). These decisions depends on the agent knowledge ($\text{price}_a, \text{warranty}_a, \text{price}_b, \text{warranty}_b$). The user also provides, through the GUI, users' preferences such that cost is important.

The theory corresponding to the influence diagram will contain for instance the two following rules:

$r_{012} : \text{provision} \leftarrow \text{cost}, \text{qos}$

$r_{01} : \text{provision} \leftarrow \text{cost}$

expressing that achieving cost and qos is ideally required to reach provision (cf r_{012}), but this can be relaxed: according to r_{01} , achieving the goal cost is enough to reach provision. Priorities amongst goals are represented by means of priorities over rules, $r_{012} \mathcal{P} r_{01}$.

In this way, our AF shifts from the goals and the preferences provided by the user to an abstract representation of atomic services (or composite services, as appropriate). For instance, an admissible argument concludes that the goal related to the provision of the service S is reached if we suppose that the price of the service S_b is low. This is then turned into a concrete representation (choice of x fulfilling the constraint) by the social reasoning.

Social Reasoning

We consider our AF to perform the social reasoning which is about the concrete instances of services which can be provided/requested. Decisions are made according to the individual statements and the social statements, i.e. the user's requirements or competencies about the service providers, the alternative concrete services, and preferences over them. The social statements are exchanged during dialogues.

The *commitments* are internal data structures which contain propositional/action social obligations involving the agent, namely with the agent being either the debtor or the creditor. Concretely, the commitments may contain the concrete representation of atomic or composite services provided by the interlocutors and the representation of the partners exchanged during the dialogues, while the *social theory* contains the concrete representation of atomic or composite services provided by the agent. Moreover, the lat-

ter contains preferences about the services and the partners. The selection (resp. the suggestion) of concrete services is made according to the user's requirements (resp. competencies) about the alternative concrete services, the information about the partners, and preferences over them.

In the influence diagram representing the negotiation problem of S_b from A1's viewpoint (cf bottom of Fig. 1), the evaluation of the contract (contract) depends on the provision of the service (provision) as considered by the individual reasoning and depends also on the supplier (supplier). The evaluation of the supplier is influenced by knowledge about customer testimonials (testimonials(x, v)) depending on the number of previous collaboration with these suppliers (previous(x, n)) and the average satisfaction in these collaboration (satisfaction(x, v)). The commitments of A1 include commitments involving A1: either A1 is the creditor of the commitment, for instance $\text{commit}(\text{Bob}, S_b(e))$ is added to the commitments when Bob suggests it; or A1 is the debtor of the commitment, for instance $\text{commit}(\text{A1}, S_b(e))$ is added to the commitments when A1 accepts it (see the next section).

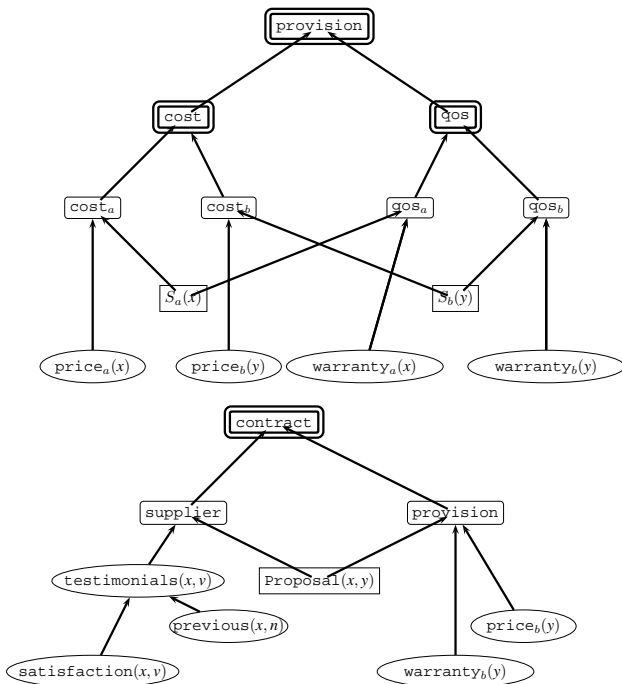


Figure 1: Influence diagram to structure the decision (top) and the negotiation (bottom)

In this way, our AF reasons, takes some decisions, and justifies them during the dialogues. For instance, the argument concluding that the goal related to the cost of the service S_b is reached since the price of $S_b(c)$ is low, is useful for A1 to justify its choice in front of Bob.

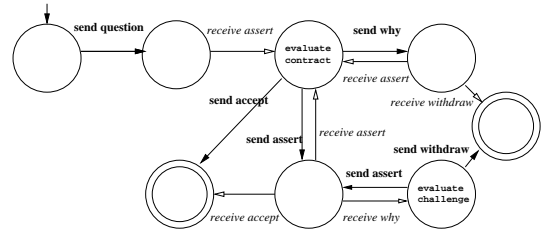


Figure 2: Negotiation protocol for the requester

Social Interaction

Our agent drives the communications by the adherence to protocols. Decisions required to conduct the interaction are provided by the social reasoning which uses a boot strap mechanism that initiates the required protocol, the role the agent will play in that protocol, and the other participants. The protocol engine determines the appropriate message to be sent given those parameters. When there is a decision to be made either between the choice of two locutions (e.g. an accept or a reject) to be sent or the instantiation of the content of the locution (e.g. the definition of a proposal), the protocol engine uses a precondition mechanism to prompt the social reasoning. Upon the satisfaction of the precondition, the protocol engine sends the locution to the outgoing message queue. A similar mechanism is used for incoming messages. If it is necessary to update the commitments of the agent, this can be done with the post condition mechanism which operates in a similar manner.

The agents utter messages to exchange goals, decisions, and knowledge. The syntax of messages is in conformance with a common communication language. We assume that each message has an identifier, M_k ; is uttered by a speaker (S_k); is addressed to a hearer (H_k); responds to a message with identifier R_k ; is characterised by a speech act A_k composed of a locution and a content. The locution is one of the following: question, assert, accept, why, withdraw (see Table 1 below for examples). The content is a triple consisting of: a goal G_k , a decision D_k , and a knowledge K_k . We will use θ to denote that no goal is given and \emptyset to denote that no knowledge is provided.

Fig. 2 depicted our negotiation protocol from the requester viewpoint with the help of a deterministic finite-state automaton. The choice of locutions to send is dependant on the way the social reasoning fulfills preconditions. For example, the outcome of evaluate contract by the social reasoning will dictate to the protocol engine whether it sends accept, assert or why. According to the corresponding rules, the commitments are updated when an assertion is received. If an admissible contract have been already suggested, then the speech act is an accept. If a new admissible contract is found, then the speech act is an assert. Otherwise the speech act is a why.

Table 1 depicts the speech acts exchanged between A1 and Bob playing a negotiation dialogue in the step 6 of the e-procurement process. They attempt to come to an agreement on the contract for the provision of a service S_b to reach the common goal good_deal . A contract is a tu-

M_k	S_k	H_k	A_k	R_k
M_0	Al	Bob	question(goodDeal, (cid, Bob, Al, $S_b(r)$, [price(x, w), warranty(x, w)]), 0)	\emptyset
M_1	Bob	Al	assert(goodDeal, (c1, Bob, Al, $S_b(d)$, [price(d, high), warranty(d, low)]), 0)	M_0
M_2	Al	Bob	assert(goodDeal, (c2, Bob, Al, $S_b(c)$, [price(c, low), warranty(c, high)]), 0)	M_1
M_3	Bob	Al	why(goodDeal, (c2, Bob, Al, $S_b(c)$, [price(c, low), warranty(c, high)]), 0)	M_2
M_4	Al	Bob	assert(cost _{Al} , (c2, Bob, Al, $S_b(c)$, [price(c, low), warranty(c, high)]), 0)	M_3
M_5	Bob	Al	assert(goodDeal, (c3, Bob, Al, $S_b(e)$, [price(e, low), warranty(e, low)]), 0)	M_4
M_6	Al	Bob	accept(goodDeal, (c3, Bob, Al, $S_b(e)$, [price(e, low), warranty(e, low)]), 0)	M_5

Table 1: Negotiation dialogue

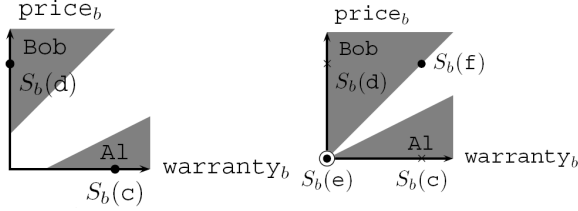


Figure 3: Acceptability space of participants after the message M_3 (left) and M_6 (right)

ple $\langle cid, deb, cred, s, terms \rangle$ where cid is the contract identifier, deb is the agent providing the service, $cred$ is the agent requesting the service. A contract is concerned by the provision of a service provided that the list of terms & conditions (denoted $terms$) are satisfied. With the message M_1 , Bob informs Al that it finds out that the terms & conditions of the contract for the provision of the service $S_b(d)$ are justified with respect to the common goal ($good_deal$). However, Al does not find $S_b(d)$ justified and he proposes $S_b(c)$. Since none of these proposals have been jointly accepted, they should not be considered in the following of the negotiation. Bob attempts to determine the reasons for Al's choice (cf M_3) which is the cost (rather than the quality of service). Given Al's response in M_4 , Bob includes the goal provided by Al. Therefore, it finds amongst the other solutions ($S_b(e)$ and $S_b(f)$) the one preferred by Al ($S_b(e)$) and suggest it (M_5). Finally, Al communicates his agreement with the help of an $accept$ (M_6) which closes the dialogue.

The $terms$ considered for the evaluation of the contract about S_b during the negotiation are represented at the two axis of the plot in Fig. 3. The acceptability space of the two participants is represented by shaded areas and depends on the price (y-axis) and the warranty (x-axis). Four points reflect the combinations of values: $S_b(c)$, $S_b(d)$, $S_b(e)$, and $S_b(f)$. After the message M_3 (cf left of Fig. 3), Bob only finds $S_b(d)$ justified and Al only finds $S_b(c)$ justified. After the negotiation dialogue (cf right of Fig. 3), the acceptability spaces of the two agents have shifted to a point where there is a common solution, $S_b(e)$. We can notice that the influence of Al on Bob avoids to explore the alternative $S_b(f)$ which is not justified from Al's viewpoint. The agents are able to influence each other, through argumentation.

Conclusions

We have presented a model of argumentative agents which, contrary to (Amgoud, Dimopoulos, & Moraitis 2007), have been implemented and tested with real-world use cases. Our proposal is not the first attempt in this direction. For in-

stance, the framework proposed by (Kakas & Moraitis 2006) allows effective argumentation-based negotiation. With respect to the latter, we have used the argumentation framework for decision-making proposed in (Morge & Mancarella 2007) which introduces multi-criteria techniques for the decision making. We incorporate abduction on missing information as suggested by (Kakas & Moraitis 2003).

In this paper, we have described an argumentative agent model to automatise the selection and composition of services. For this purpose we have provided an AF for decision-making to perform the individual reasoning and the social reasoning. The individual reasoning is about how to achieve its individualistic goals and the social reasoning is about common goal solving through collaboration. In order to valid this approach, we use the multiagent platform GOLEM (Bromuri & Stathis 2007) for the deployment of our agents.

Acknowledgements

This work is supported by the Sixth Framework IST programme of the EC, under the 035200 ARGUGRID project.

References

- Amgoud, L.; Dimopoulos, Y.; and Moraitis, P. 2007. A unified and general framework for argumentation-based negotiation. In *Proc. of AAMAS*, 963–970.
- Bromuri, S., and Stathis, K. 2007. Situating cognitive agents in GOLEM. In *Proc. of the EEMMAS*, 76–93.
- Dung, P. M.; Mancarella, P.; and Toni, F. 2007. Computing ideal sceptical argumentation. *Artif. Intell. Special Issue on Argumentation* 171(10-15):642–674.
- Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.* 77(2):321–357.
- Gartner, D., and Toni, F. 2007. CaSAPI: a system for credulous and sceptical argumentation. In *Proc. of ArgNMR*, 80–95.
- Hendler, J. 2001. Agents and the semantic web. *IEEE Intelligent systems* 16(2):30–37.
- Kakas, A., and Moraitis, P. 2003. Argumentative-based decision-making for autonomous agents. In *Proc of AAMAS*, 883–890. ACM Press.
- Kakas, A., and Moraitis, P. 2006. Adaptive agent negotiation via argumentation. In *Proc. of AAMAS*, 384–391.
- Kakas, A. C.; Mancarella, P.; Sadri, F.; Stathis, K.; and Toni, F. 2004. The KGP model of agency. In *Proc. of ECAI*, 33–37.
- Morge, M., and Mancarella, P. 2007. The hedgehog and the fox. An argumentation-based decision support system. In *Proc. of ARGMAAS*, 55–68.
- Stournaras, T., ed. 2007. *eBusiness application scenarios*. Deliverable document D1.2 ARGUGRID.
- Vreeswijk, G. 1997. Abstract argumentation systems. *Artif. Intell.* 90(1-2):225–279.