DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN
UNIVERSITAT POLITÈCNICA DE VALÈNCIA

P.O. Box: 22012 E-46071 Valencia (SPAIN)

Departamento de Sistemas Informáticos y Computación

Technical Report

Ref. No.:
Pages: 21
Title: Case-based Argumentation Framework. Strategies
Author(s): Stella Heras and Vicente Botti and Vicente Julián
Date: June 24, 2011
Keywords: Argumentation, Case-Based Reasoning, Multi-Agent Systems

V° B°
Vicente Botti Stella Heras

1 Introduction

Nowadays, most large scale computer systems are viewed as service-oriented systems which software components perform business transactions by providing and consuming services. These components are often implemented as agents in a Multi-Agent System (MAS). Agents are able to reason autonomously, represent human interests and preferences and group together to form societies that link them via dependency relations. In this societies, agents perform complex tasks that require different levels of intelligence and give rise to interactions among them. From these interactions, conflicts of opinion can arise, specially when MAS become adaptive and open (with heterogeneous agents dynamically entering in or leaving the system). Therefore, software agents willing to participate in this type of systems will require to include extra capabilities to explicitly represent and generate *agreements* [24].

Argumentation provides a natural means of dealing with conflicts of interest and opinion. Agents can reach agreements by engaging in argumentation dialogues with their opponents in a discussion. Therefore, the study of argumentation in MAS has gained a growing interest [20, Part III]. Case-Based Reasoning (CBR) [1] is another research area where the argumentation theory has produced a wide history of successful applications. According to the CBR methodology, a new problem can be solved by searching in a case-base for similar precedents and adapting their solutions to fit the current problem. This reasoning methodology has a high resemblance with the way by which people argue about their positions, trying to justify them on the basis of past experiences. The work done in the eighties about legal CBR fostered the argumentation research in the AI community [22]. From then on, the good results of CBR systems in argumentation domains suggest that this type of reasoning is suitable to manage argumentation processes [10].

On top of the simpler capacity to argue and reach agreements, agents can take advantage of previous argumentation experiences to follow dialogue strategies and easily persuade other agents to accept their opinions. However, the social dependencies of agents and their preferences over values (goods that agents want to promote or demote) determine the way in which they can argue with other agents. For instance, agents that represent workers in a company are not as willing to accept arguments from equals as they are from superiors (due to a power dependency relation between superiors and other workers). Similarly, an agent representing a trade union does not argue (for instance, to increase salaries and improve the workers economy) in the same way as if it would be defending its individual values (for instance, to improve its reputation within the company).

Starting from the idea that the social context of agents determines the way in which agents can argue and reach agreements, this context should have a decisive influence in the computational representation of arguments, in the argument management process and in the way agents develop strategies to argue with other agents. While work on argument generation and evaluation has received much attention, the strategic use of arguments has received little attention in the literature [15]. Our point of view is that CBR can be very useful to manage argumentation in open MAS and devise argumentation strategies based on previous argumentation experiences. To demonstrate the foundations of this suggestion, this report presents the work that we have done to develop case-based argumentation strategies in agent societies.

First, Section 2 briefly presents the case-based argumentation framework that we have developed to represent and manage arguments in agent societies. The framework takes into account the agents' social context in the way agents can argue. Thus, social dependencies between agents and their values are also considered. Also, agents implementing this framework will be able to engage in argumentation dialogues following different dialogue strategies. With these strategies, agents will be able to select the most appropriate argument to bring about their desired outcome of the dialogue. Then, Section 3 shows the case-based dialogue strategies proposed in this work. In Section 4 the framework is implemented in a real customer support application and the different dialogue strategies that agents can follow are evaluated. Section 5 analyses related work and compares our approach with it. Finally, Section 6 summarises the main contributions of this work.

2 Case-based Argumentation Framework

In [9, Chapter 3] a case-based argumentation framework for agent societies has been proposed. In this section, we briefly introduce the elements of the framework that allow us to develop the case-based argumentation strategies presented in this report. Concretely, in our framework we propose three types of knowledge resources that the agents can use to generate, select and evaluate arguments following different argumentation strategies:

A case-base with domain-cases that represent previous problems and their solutions. Agents can use this knowledge resource to generate their positions in a dialogue and arguments to support them. Also, the acquisition of new domain-cases increases the knowledge of agents about the domain under discussion.

A case-base with argument-cases that store previous argumentation experiences and their final outcome. Argument-cases have three main objectives: they can be used by agents 1) to generate new arguments; 2) to strategically select the best position to put forward in view of past argumentation experiences; and 3) to store the new argumentation knowledge gained in each agreement process, improving the agents' argumentation skills.

A database of Argumentation schemes with a set of argumentation schemes [27], which represent stereotyped patterns of common reasoning in the application domain where the framework is implemented. An argumentation scheme consists of a set of premises and a conclusion that is presumed to follow from them. Also, each argumentation scheme has associated a set of *critical questions* that represent potential attacks to the conclusion supported by the scheme. The concrete argumentation schemes to be used depend on the application domain.

The structure of domain-cases and the concrete set of argumentation schemes that an argumentation system that implements our framework has depends on the application domain. Argument-cases are the main structure that we use to computationally represent arguments in agent societies. In addition, their structure is generic and domain-independent. Table 1 shows the generic structure of an argument-case.

<table>
<tr><td rowspan="11">PROBLEM</td><td>Domain Context</td><td colspan="2">[Premises]*</td></tr>
<tr><td rowspan="10">Social Context</td><td rowspan="3">Proponent</td><td>ID</td></tr>
<tr><td>Role</td></tr>
<tr><td>ValPref</td></tr>
<tr><td rowspan="3">Opponent</td><td>ID</td></tr>
<tr><td>Role</td></tr>
<tr><td>ValPref</td></tr>
<tr><td rowspan="3">Group</td><td>ID</td></tr>
<tr><td>Role</td></tr>
<tr><td>ValPref</td></tr>
<tr><td colspan="2">Dependency Relation</td></tr>
<tr><td rowspan="6">SOLUTION</td><td colspan="3">Conclusion</td></tr>
<tr><td colspan="3">Value</td></tr>
<tr><td colspan="3">Acceptability Status</td></tr>
<tr><td rowspan="3">Received Attacks</td><td colspan="2">[Critical Questions]*</td></tr>
<tr><td colspan="2">[Distinguishing Premises]*</td></tr>
<tr><td colspan="2">[Counter Examples]*</td></tr>
<tr><td rowspan="3">JUSTIFICATION</td><td colspan="3">[Cases]*</td></tr>
<tr><td colspan="3">[Argumentation Schemes]*</td></tr>
<tr><td colspan="3">Associated Dialogue Graphs</td></tr>
</table>

Table 1: Structure of an Argument-Case.

Argument-cases have the three possible types of components that usual cases of CBR systems have: the description of the state of the world when the case was stored (*Problem*); the solution of the case (*Conclusion*); and the explanation of the process that gave rise to this conclusion (*Justification*).

The problem description has a *domain context* that consists of the *premises* that characterise the argument. In addition, if we want to store an argument and use it to generate a persuasive argument in the future, the features that characterise its *social context* must also be kept. The social context of the argument-case includes information about the *proponent* and the *opponent* of the argument and about their *group*. Moreover, we also store the preferences (*ValPref*) of each agent or group over the set of *values* pre-defined in the system. Finally, the *dependency relation* between the proponent's and the opponent's roles is also stored. In our framework, we consider three types of dependency relations as defined in [8]: *Power*, when an agent has to accept a request from other agent because of some pre-defined domination relationship between them; *Authorisation*, when an agent has signed a contract with other agent to provide it with a service and hence, the contractor agent is able to impose its authority over the contracted agent and *Charity*, when an agent is willing to answer a request from other agent without being obliged to do so.

In the solution part, the *conclusion* of the case, the *value* promoted, and the *acceptability status* of the argument at the end of the dialogue are stored. The last feature shows if the argument was deemed *acceptable*, *unacceptable* or *undecided* in view of the other arguments that were put forward in the agreement process. In addition, the conclusion part includes information about the possible *attacks* that the argument received during the process. These attacks could represent the justification for an argument to be deemed unacceptable or else reinforce the persuasive power of an argument that, despite being attacked, was finally accepted. Concretely, arguments in our framework can be attacked by putting forward *distinguishing premises* or *counter-examples* to them, as proposed in [6], and also by questioning the validity of the conclusion drawn from an argumentation scheme by instantiating a *critical question*.

Definition 2.1 (Distinguishing Premise). *A distinguishing premise x with respect to a problem P between two cases* $c_1, c_2 \in C$ *is defined as:* $\exists x \in c_1 \wedge \not\exists x \in P / \exists x \in c_2 \wedge value_{c_1}(x) \neq value_{c_2}(x)$ *or else,* $\exists x \in c_1 \wedge \exists x \in P / value_{c_1}(x) = value_P(x) \wedge \nexists x \in c_2$*, where* $P \subseteq F$*,* $x \in F$ *and* $c_1, c_2 \in C$.

That is a premise that does not appear in the description of the problem to solve and has different values for two cases or a premise that appears in the problem description and does not appear in one of the cases.

Definition 2.2 (Counter-Example). *A counter-example for a case is a previous domain-case or an argument-case that was deemed acceptable, where the problem description of the counter-example matches the current problem to solve and also subsumes the problem description of the case, but proposing a different solution.*

Definition 2.3. *A critical question is a question associated to an argumentation scheme that represents a potential way in which the conclusion drawn from the scheme can be attacked.*

Therefore, if the opponent asks a critical question, the argument that supports this argumentation scheme remains temporally rebutted until the question is conveniently answered. This characteristic of argumentation schemes makes them very suitable to devise ways of attack the conclusions drawn from other agents.

Finally, the justification part of an argument-case stores the information about the knowledge resources that were used to generate the argument represented by the argument-case (the set of domain-cases and argument-cases). In addition, the justification of each argument-case has a *dialogue-graph* (or several) associated, which represents the dialogue where the argument was proposed. In this way, the sequence of arguments that were put forward in a dialogue is represented, storing the complete conversation as a directed graph that links argument-cases. This graph can be used later to improve the efficiency in an argumentation dialogue in view of a similar dialogue that was held in the past.

As pointed out before, in our framework agents can generate arguments from previous cases (domain-cases and argument-cases) and from argumentation schemes. However, note that the fact that a proponent agent uses one or several knowledge resources to generate an argument does not imply that it has to show all this information to its opponent. The argument-cases of the agents' argumentation systems and the structure of the actual arguments that are interchanged between agents is not the same. Thus, arguments that agents interchange are defined as tuples of the form:

Definition 2.4 (Argument)**.** *$Arg = \{\phi, v, < S >\}$, where ϕ is the conclusion of the argument, v is the value that the agent wants to promote with it and $< S >$ is a set of elements that support the argument (support set).*

This support set can consist of different elements, depending on the argument purpose. On one hand, if the argument provides a potential solution for a problem, the support set is the set of features (*premises*) that represent the context of the domain where the argument has been proposed (those premises that match the problem to solve and other extra premises that do not appear in the description of this problem but that have been also considered to draw the conclusion of the argument) and optionally, any knowledge resource used by the proponent to generate the argument (*domain-cases*, *argument-cases* or *argumentation schemes*). On the other hand, if the argument attacks the argument of an opponent, the support set can also include any of the allowed attacks in our framework (*critical questions*, *distinguishing premises* or *counter-examples*).

Next section explains how we can make use of the elements of this case-based argumentation framework to devise heuristic argumentation strategies.

3 Case-based Dialogue Strategies

In this section we propose several dialogue strategies for different types of agent profiles (agent attitudes) in our case-based argumentation framework. In doing so, we are closer to the heuristic approach of [2] rather than to game theoretic one. The reasons behind this decision will be discussed in Section 5.

In each step of the dialogue of our case-based argumentation framework an agent can choose a specific locution and a content for it depending on its profile and strategy. The specific locutions and dialogue rules that agents follow to make such decisions is out of the scope of this report and is presented in the dialogue protocol described in [9, Chapter 4]. Also, the concrete reasoning process and algorithms that agents follow to generate, select and evaluate arguments from their knowledge resources is explained in [9, Chapter 3]. Here, assuming that L represents the set of available locutions of the protocol that agents use to communicate, let us suppose that the function *Replies* returns for each locution the legal replies to it:

$$Replies : L \rightarrow 2^L$$

Then, assuming that D represents the set of well-formed formulae in the logical language used in the framework to represent arguments and knowledge resources [9, Chapter 3], the function *Content* returns for a given locution, the set of possible contents:

$$Content : L \rightarrow 2^D$$

In each step of the argumentation dialogue, agents exchange moves.

Definition 3.1 (Move)**.** *A move is a pair (l, ϕ), where $l \in L$ and $\phi \in$ Content(l).*

Thus, the strategy problem is formalised as in [2]:

Definition 3.2 (Strategy Problem)**.** *Let (l, ϕ) be the current move in a dialogue. What is the next move (l', ϕ') to utter such that l' $\in$ Replies(l)?*

The answer for this question implies to find the best locution and content that the agent can utter in each step of the dialogue, given the profile of the agent and its knowledge resources. Therefore, a dialogue strategy is defined as follows:

Definition 3.3 (Dialogue Strategy)**.** *A dialogue strategy is defined as a function S: $2^{L \times D} \rightarrow L \times D$ where (l, ϕ) $\in L \times D$.*

Given a move $m=(l, \phi)$, $S(m) = m'$ such that $m' = (l', \phi')$ is the best move that an agent can utter in the next step of the dialogue taking into account its profile and knowledge. In our case-based argumentation framework, agents select the best locution to bring up depending on their *profile* and the content of this locution depending on the knowledge that they have in their *knowledge resources* and the tactic that they follow.

We consider the following agent profiles [5]:

- Agreeable: accept whenever possible.
- Disagreeable: only accept when there is no reason not to.
- Open-minded: only attack when necessary.
- Argumentative: attack whenever possible.
- Elephant's child: challenge whenever possible.

An agreeable agent will initially accept positions and arguments from peers (other agents that it has a charity dependency relation with them) whenever is possible. This means that it will accept any position that is in the list of its potential positions (even if it is not ranked as the most suitable) and it will accept any argument from a peer if it does not have a counter-argument, a distinguishing premise or a critical question that attack it. Therefore, if the agent cannot rebut an argument from a peer, it will accept it and also its associated position. Agreeable agents do not challenge positions of other agents, but just try to defend theirs if attacked. In the case that an agreeable agent cannot generate a position, it does not participate in the dialogue.

A disagreeable agent will initially accept the position of a peer if it is ranked first in its list of potential positions. Regarding arguments, this type of agent will try to generate an attack to any argument that it receives from other agents. If it is not able to generate such an attack, the agent will accept the argument of its peer, but it still will not accept the peer's position. Disagreeable agents do not challenge positions of other agents, but just try to defend theirs if attacked. In the case that a disagreeable agent cannot generate a position, it does not participate in the dialogue.

An open-minded agent challenges different positions of other peers. Also, it waits for challenges from other peers and will try to rebut their attack arguments. If a peer wins the discussion, this type of agent accepts its argument and its associated position. If it cannot generate positions, it does not engage in the dialogue.

An argumentative agent will not initially accept any position from a peer. This type of agent will challenge positions of other peers when they are different from its position, even if they appear in its list of potential positions to propose. Also, it will try to generate an answer for any attack that it receives, but opposite to open-minded agents, argumentative agents do not accept the position of the peer that generated the attack if the last wins the debate. If an argumentative agent cannot generate positions, it will not participate in the dialogue.

An elephant's child agent will always challenge the positions of other peers (even if they have proposed the same position than it or if it cannot generate positions). If it can generate attacks, it will put forward them to rebut the arguments of other agents, but if they win the debate, this type of agent does not accept their positions. In fact, the only way an elephant's child agent accepts the position of other agent is when it attacks this position and the attacked agent wins the debate.

Independently of their profile, agents will accept arguments from other agents that have a power or authorisation dependency relation over them. Recall that in any case the acceptance of an argument is subjected to the defeat relation defined in the argumentation framework [9, Chapter 3]. Table 2 summarises the behaviour of these agent profiles.

The legends of the columns of Table 2 are the following:

- **ASP (Accept Same Position):** Initially accept the position of a peer that matches the agent's current position.

	ASP	APL	CSP	CPL	CDP	AAP	CP∅
Agreeable	√	√					
Disagreeable	√						
Open-Minded					√	√	
Argumentative				√	√		
Elephant's Child			√	√	√		√

Table 2: Agents' Profiles

- **APL (Accept Position in List):** Initially accept the position of a peer that is in the agent's list of potential positions (although not ranked as the most suitable position to propose).
- **CSP (Challenge Same Position):** Challenge the position of a peer that has proposed the same position than the agent's one.
- **CPL (Challenge Position in List):** Challenge the position of a peer that has proposed a position that is in the agent's list of potential positions (although not ranked as the most suitable position to propose).
- **CDP (Challenge Different Position):** Challenge the position of peer that has proposed a different position (and this position is not in the agent's list of potential positions to propose).
- **AAP (Accept Attacked Position):** When the agent has held a discussion with other agent and the later wins the debate, accept its position.
- **CP∅ (Challenge other Positions if no position can be generated).**

As pointed out before, depending on its profile, the agent will choose the next locution to put forward on the dialogue. For instance, let us assume that an agent a_i has proposed a position q to solve the problem p under discussion, an agreeable agent a_j has entered in the dialogue and proposed a position q' and a_i has challenged the position of a_j. In this case, the agreeable agent a_j will try to generate a support argument for its position by searching its domain and argument-cases case-bases. Then, among the potential arguments that a_j could generate, it has to select one to support the position. This implies to select the content of the locution to assert the support argument. To make this selection, agents can use several tactics, which consist on assigning more or less weight to the elements of a *support factor* used to select positions and arguments (see [9, Chapter 3] for a detailed explanation about the argument generation and selection process). This support factor estimates how suitable a current argument is in view of the suitability of similar arguments put forward in previous argumentation dialogues.

Actually, what the agent does is to decide which argument is most suitable in view of its past experience, stored in the form of argument-cases. We consider the parameters shown in the following formulas as criteria for making such decision. In the formulas, $argC$ is the number of argument-cases from the set arg with the same conclusion than the current argument, $argAccC$ are those in $argC$ that were deemed acceptable, $argAccCAtt$ are those in $argAccC$ that were attacked, $minAtt$ and $maxAtt$ are the minimum and maximum number of attacks received by any argument generated, $minS$ and $maxS$ are the minimum and maximum number of steps from any retrieved argument-case to the last node of its dialogue graph and $minKr$ and $maxKr$ are the minimum and maximum number of knowledge resources used to generate any argument.

- **Persuasiveness Degree (PD):** is a value that represents the expected persuasive power of an argument by checking how persuasive an argument-case with the same problem description and conclusion was in the past. To compute this degree, the number $argAccC$ of argument-cases that were deemed acceptable out of the total number of retrieved argument-cases $argC$ with the same problem description and conclusion than the current argument is calculated:

$$PD = \begin{cases} 0, & if\ argC = \emptyset \\ \dfrac{argAccC}{argC}, & otherwise \end{cases} \tag{1}$$

with $argAccC$, $argC \in N$ and $PD \in [0, 1]$, from less to more persuasive power.

- **Support Degree (SD):** is a value that provides an estimation of the probability that the conclusion of the current argument was acceptable at the end of the dialogue. It is based on the number of argument-cases $argAccC$ with the same problem description and conclusion that where deemed acceptable out of the total number of argument-cases arg retrieved.

$$SD = \begin{cases} 0, & if\ arg = \emptyset \\ \dfrac{argAccC}{arg}, & otherwise \end{cases} \tag{2}$$

with $argAccC, arg \in N$ and $SD \in [0, 1]$ from less to more support degree.

- **Risk Degree (RD):** is a value that estimates the risk for an argument to be attacked in view of the attacks received for an argument(s) with the same problem description and conclusion in the past. It is based on the number of argument-cases $argAccCAtt$ that were attacked out of the total number of $argAccC$ argument-cases with the same problem description and conclusion retrieved that were deemed acceptable.

$$RD = \begin{cases} 0, & if\ argC = \emptyset \\ \dfrac{argAccCAtt}{argAccC}, & otherwise \end{cases} \tag{3}$$

with $argAccCAtt$, $argC \in N$ and $RD \in [0, 1]$, from less to more risk of attack.

- **Attack degree (AD):** is a value that provides an estimation of the number of attacks att received by a similar argument(s) in the past. To compute this degree, the set of arguments with the same problem description that were deemed acceptable is retrieved. Then, this set is separated in several subsets, one for each different conclusion. The sets whose conclusion match with the conclusions of the arguments to assess are considered, while the other sets are discarded. Thus, we have a set of argument-cases for each different potential argument that we want to evaluate. For each argument-case in each set, the number of attacks received is computed (the number of critical questions, distinguishing premises and counter-examples received). Then, for each set of argument-cases, the average number of attacks received is computed. The attack degree of each argument is calculated by a linear transformation:

$$AD = \begin{cases} 0, & if\ maxAtt = minAtt \\ \dfrac{att - minAtt}{maxAtt - minAtt}, & otherwise \end{cases} \tag{4}$$

with $minAtt$, $maxAtt$, $att \in N$ and $AD \in [0, 1]$ from less to more degree of attack.

- **Efficiency degree (ED):** is a value that provides an estimation of the number of steps that took to reach an agreement posing a similar argument in the past. It is based on the depth n from the node representing a similar argument-case retrieved to the node representing the conclusion in the dialogue graphs associated to it. To compute this degree, the same process to create the subsets of argument-cases than in the above degree is performed. Then, for each argument-case in each subset, the number of dialogue steps from the node that represents this argument-case to the end of dialogue is computed. Also, the average number of steps per subset is calculated. Finally, the efficiency degree of each argument is calculated by a linear transformation:

$$ED = \begin{cases} 0, & if\ maxS = minS \\ 1 - \dfrac{n - minS}{maxS - minS}, & otherwise \end{cases} \tag{5}$$

with $minS$, $maxS$, $n \in N$ and $ED \in [0, 1]$ from less to more efficiency.

- **Explanatory Power (EP):** is a value that represents the number of pieces of information that each argument covers. It is based on the number kr of knowledge resources were used to generate each argument. To compute this number, the same process to create the subsets of argument-cases than in the above degrees is performed. Then, for each argument-case in each set, the number of knowledge resources in the justification part is computed (the number of domain-cases, argument-cases and argumentation schemes). Then, for each set of argument-cases, the average number of knowledge resources used is computed. The explanatory power of each argument is calculated by a linear transformation:

$$EP = \begin{cases} 0, & if\ maxKr = minKr \\ \dfrac{kr - minKr}{maxKr - minKr}, & otherwise \end{cases} \tag{6}$$

with $minKr$, $maxKr$, $kr \in N$ and $EP \in [0, 1]$ from less to more explanatory power.

Finally, the support factor of a new argument is computed by the formula:

$$\begin{aligned} SF =& ((w_{PD} * PD + w_{SD} * SD + w_{RD} * (1 - RD) \\ &+ w_{AD} * (1 - AD) + w_{ED} * ED + w_{EP} * EP)) \end{aligned} \tag{7}$$

where $w_i \in [0, 1]$, $\sum w_i = 1$ are weight values that allow the agent to give more or less importance to each decision criteria. Thus, depending on this weights, an agent can follow the following tactics:

- Persuasive Tactic: the agent selects such arguments which similar argument-cases were more persuasive in the past (have more persuasiveness degree).
- Maximise-Support Tactic: the agent selects such arguments that have higher probability of being accepted at the end of the dialogue (their similar argument-cases have more support degree).
- Minimise-Risk Strategy: the agent selects such arguments that have a lower probability of being attacked (their similar argument-cases have less risk degree).
- Minimise-Attack Tactic: the agent selects such arguments that have received a lower number of attacks in the past (their similar argument-cases have less attack degree).
- Maximise-Efficiency Tactic: the agent selects such arguments that lead to shorter argumentation dialogues (their similar argument-cases have higher efficiency degree).
- Explanatory Tactic: the agent selects such arguments that cover a bigger number of cases or argumentation schemes. That is, such arguments that are similar to argument-cases that have more justification elements (more cases or argumentation schemes in the justification part).

In this section, we have explained the different strategies that agents of our case-based argumentation framework can follow during the dialogue. Now, the following section implements these strategies in a real application domain and analyses the results achieved.

4 Evaluation

In this section, we evaluate the performance of the case-based argumentation framework presented in this report by running a set of empirical tests. With this objective, the framework has been implemented in the domain of a customer support application. Concretely, we consider a society of agents that act in behalf of a group of technicians that must solve problems in a Technology Management Centre (TMC). TMCs are entities which control every process implicated in the provision of technological and customer support services to private or public organisations. Usually, TMCs are managed by a private company that communicates with its customers via a call centre. This kind of centres allow customers to obtain general information, purchase products or lodge a complaint. They can also efficiently communicate public administrations with citizens. In a call centre, there are a number of technicians attending to a big amount of calls with different objectives –sales, marketing, customer service, technical support and any business or administration activity–. The call centre technicians have computers provided with a helpdesk software and phone terminals connected to a telephone switchboard that manages and balances the calls among technicians. The current implementation is based in previous work that deployed a case-based multi-agent system in a real TCM [11]. This system was implemented and is currently used by the TCM company. In the original implementation, agents were allowed to use their case-bases to provide experience-based customer support. In this work, the system has been enhanced by allowing agents to argue about the best way of solving the incidences that the call centre receives.

Therefore, we consider a society composed by call-centre technicians with two possible roles, *operator* and *expert*. Operators form groups that must solve the problems that the call centre receives. Experts are specialised operators that have case-bases with knowledge about the suitable solutions to provide for specific problems. Therefore, the dependency relations in this society establish that experts have a power relation over operators and that technicians with the same role have a charity relation among them.

In this application domain we assume that each technician has a helpdesk application to manage the big amount of information that processes the call centre. The basic functions of this helpdesk are the following:

- To register the ticket information: customer data, entry channel and related project, which identifies the customer and the specific service that is being provided.
- To track each ticket and to scale it from one technician to a more specialised one or to a different technician in the same level.
- To warn when the maximum time to solve an incidence is about to expire.
- To provide a solution for the ticket. This means to generate an own position or to ask for help to the members of a group.

In addition, this helpdesk would implement an argumentation module to solve each ticket as proposed in our framework. Hence, we assume the complex case where a ticket must be solved by a group of agents representing technicians that argue to reach an agreement over the best solution to apply. Each agent has its own knowledge resources (acceded via his helpdesk) to generate a solution for the ticket. The data-flow for the problem-solving process (or argumentation process) to solve each ticket is the following:

1. The system presents a group of technicians with a new ticket to solve.
2. If possible, each technician generates his own position by using the argumentation module. This module supports the argumentation framework proposed in this report.
3. All technicians that are willing to participate in the argumentation process are aware of the positions proposed in each moment.
4. The technicians argue to reach an agreement over the most suitable solution by following a deliberation dialogue controlled by the proposed dialogue game protocol.
5. The best solution is proposed to the user and feedback is provided and registered by each technician helpdesk.

The helpdesk of each technician is provided with a case-based reasoning engine that helps them to solve the ticket. The new argumentation module will allow different technicians to reach agreements over the best solution to apply in each specific situation. In this example application, we assume that the most efficient technicians are acknowledged and rewarded by the company. Therefore, each technician follows a *persuasion* dialogue with their partners, trying to convince them to accept its solution as the best way to solve the ticket received, while observing the common objective of providing the best solution for a ticket from its point of view.

For the tests, a real database of 200 tickets solved in the past is used as domain knowledge. Translating these tickets to domain-cases, we have obtained a tickets case-base with 48 cases. Despite the small size of this case-base, we have rather preferred to use actual data instead of a larger case-base with simulated data. The argument-cases case-bases of each agent are initially empty and populated with cases as the agents acquire argumentation experience in execution of the system.

To diminish the influence of random noise, for each round in each test, all results report the average and confidence interval of 48 simulation runs at a confidence level of 95%, thus using a different ticket of the tickets case-base as the problem to solve in each run. The results report the mean of the sampling distribution (the population mean) by using the formula:

$$\mu = \bar{x} \pm t * \frac{s}{\sqrt{n}} \tag{8}$$

where, $\bar{x}$ is the sample mean (the mean of the 48 experiments), t is a parameter that increases or decreases the standard error of the sample mean ($\frac{s}{\sqrt{n}}$), s is the sample standard deviation and n is the number of experiments. For small samples, say below 100, t follows the *Student's t-distribution*, which specifies certain value for the t parameter to achieve a confidence level of 95% for different sizes of population. In our case, with a population of 48 experiments the Student's t-distribution establishes a value of 2.0106 for t.

In each simulation experiment, an agent is selected randomly as initiator of the discussion. This agent has the additional function of collecting data for analysis. However, from the argumentation perspective, its behaviour is exactly the same as the rest of agents and its positions and arguments do not have any preference over others (unless there is a dependency relation that states it). The initiator agent receives one problem to solve per run. Then, it contacts its partners (the agents of its group) to report them the problem to solve. If the agents do not reach an agreement after a maximum time, the initiator chooses the most supported (the most voted) solution as the final decision (or the most frequent in case of draw). If the draw persists, the initiator makes a random choice among the most frequent solutions.

In the following tests, we have compared different argumentation strategies (as combinations of an agent profile and different tactics) that an agent can follow to argue with other agents. The argumentative profile presented in Section 3 exploits all knowledge resources of the case-based argumentation framework proposed. Other agent profiles involve different modifications of this profile. Therefore, to perform these tests, an agent (say the initiator agent) has been set to have an *argumentative* profile and to follow different tactics to argue with agents of different profiles. Also, these agents do not follow any specific tactic. By default, all agents know each other, all are in the same group and the dependency relation between them is *charity*. The values of each agent have been randomly assigned and agents know the values of their partners. Also, all agents play the role of *operator*. Different configurations of the agents' social context have been evaluated in [9, Chapter 6].

The different strategies evaluated represent the combination of the argumentative profile for the initiator with the different tactics proposed in the previous section. Recalling, these are the Persuasive Strategy (ST1), by which the agent selects such positions and arguments whose associated argument-cases were more persuasive in the past; the Maximise-Support Strategy (ST2), by which the agent selects such positions and arguments that have higher probability of being accepted at the end of the dialogue; the Minimise-Risk Strategy (ST3), by which the agent selects such position and arguments that have a lower probability of being attacked; the Minimise-Attack Strategy (ST4), by which the agent selects such positions and arguments that have received a lower number of attacks in the past; the Maximise-Efficiency Strategy (ST5), by which the agent selects such positions and arguments that lead to shorter argumentation dialogues; and the Explanatory Strategy (ST6), by which the agent selects such positions and arguments that cover a bigger number of domain-cases, argument-cases or dialogue graphs.

These strategies are evaluated by computing the agreement percentage obtained by the agents when they have a *low* (from 5 to 15 domain-cases), *medium* (from 20 to 30 domain-cases) or *high* (from 35 to 45 domain-cases) knowledge about the domain. Also, the argumentative agent has a full case-base of 20 argument-cases. Then, the percentage of agents that an argumentative agent (the initiator, for instance) is able to persuade to reach an agreement to propose its solution as the best option to solve a ticket is computed for the same settings (from low to high knowledge about the domain). To be able to follow effectively an argumentation tactic, an agent needs to have useful argument-cases to reuse these experiences for the current argumentation situation. Thus, all strategic tests report results obtained when the initiator agent has some argument-cases that match the current situation and actually uses them to select the best position and arguments to propose. The following tables show the results obtained.

Table 3 shows the percentage of times that an agreement about the best solution to apply is reached when the argumentative agent is arguing with agents that have an *agreeable* profile. Also, Table 4 shows the percentage of agreeable agents that the argumentative agent (the initiator) is able to persuade to propose its solution as the best option to solve a ticket. Agreeable agents do not challenge the positions of other agents, but only try to defend their positions if they are attacked. However, an agreeable agent accepts the position of other agent that has proposed its same position or a position that it has generated, although it has not ranked it as the best solution to apply. That means that in these cases, the agreeable agent votes the position of other agents and withdraws its own.

For low knowledge about the domain, no useful argument-cases are found and agents cannot reach an agreement about the best solution to apply to solve the ticket at hand, whatever strategy is followed. This does not means that the system is not able to propose a solution at all, but the solution proposed is not agreed by all agents. In the case of having a medium amount of knowledge about the domain, the best results are achieved for these strategies that minimise the probability or the number of potential attacks that an argument can receive (ST3 and ST4). Remember that an agreeable agent does not attack any position and if attacked, if it cannot defend itself, it just withdraw its position. Therefore, if the initiator uses arguments to generate its position and arguments that the agreeable agent cannot defeat, the initiator will have less agreeable agents competing in the dialogues and an agreement is reached more easily. However, if the agents have more knowledge about the domain, the agreeable agents increase their options to generate potential attacks to rebut the attacks that they receive, and hence, following a strategy that selects those positions and arguments that are expected to have a higher acceptability degree from the other agents (ST1) makes the initiator to be able to reach most agreements.

The results of Table 4 show again how ST3 and ST4 perform better for a medium amount of knowledge about the domain and ST1 for a high amount of knowledge. This reinforces our hypothesis that these strategies make the initiator agent to reach to a higher number of agreements by persuading other agents to accept its position as the best solution to apply for a ticket requested.

Strategy / Domain Knowledge	**ST1**	**ST2**	**ST3**	**ST4**	**ST5**	**ST6**
LOW	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
MEDIUM	25.00%	16.67%	*33.33%*	*33.33%*	0.00%	16.67%
HIGH	*63.89%*	55.56%	50.00%	25.00%	38.89%	19.44%

Table 3: Agreement Percentage with Agreeable agents.

Strategy / Domain Knowledge	**ST1**	**ST2**	**ST3**	**ST4**	**ST5**	**ST6**
LOW	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
MEDIUM	14.29%	11.90%	*22.62%*	*22.62%*	0.00%	11.90%
HIGH	*52.38%*	45.24%	38.89%	21.03%	30.95%	14.68%

Table 4: Percentage of Agreeable agents persuaded.

Table 5 shows the percentage of times that an agreement about the best solution to apply is reached when the argumentative agent is arguing with agents that have a *disagreeable* profile. In addition, Table 6 shows the percentage of agents that the argumentative agent (the initiator) is able to persuade to propose its solution as the best option to solve a ticket when it is arguing with disagreeable agents. Disagreeable agents act similarly to agreeable agents, do not challenging any position proposed by other agents, but in this case, this profile of agents only accept the position of a partner (voting it and withdrawing its own), if it exactly coincides with the position that they have proposed as the best solution to apply.

Again, low knowledge about the domain prevents the use of useful argument-cases. However, for both medium and high amounts of knowledge about the domain, the initiator agent is able to convince more agents if it follows an strategy that minimises the number of potential attacks that its position and arguments can receive, getting thus to higher agreement percentages. This results in selecting those positions that, although still serve to the initiator's agent objectives, are more similar to the positions proposed by the disagreeable agents and hence, these have less potential attacks to put forward and are more easily persuaded to reach an agreement to accept the initiator's position (as shown in Table 6). Nevertheless, disagreeable agents are difficult to convince and both agreements and agents that agree percentages are low independently of the strategy followed by the initiator.

Strategy / Domain Knowledge	**ST1**	**ST2**	**ST3**	**ST4**	**ST5**	**ST6**
LOW	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
MEDIUM	16.67%	0.00%	8.33%	*25.00%*	8.33%	16.67%
HIGH	36.11%	30.56%	11.11%	*38.89%*	30.56%	25.00%

Table 5: Agreement Percentage with Disagreeable agents.

Strategy / Domain Knowledge	**ST1**	**ST2**	**ST3**	**ST4**	**ST5**	**ST6**
LOW	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
MEDIUM	11.90%	0.00%	5.95%	*15.48%*	5.95%	11.90%
HIGH	30.56%	25.40%	9.52%	*31.35%*	25.79%	23.02%

Table 6: Percentage of Disagreeable agents persuaded.

Table 7 shows the percentage of times that an agreement about the best solution to apply is reached when the argumentative agent is arguing with agents that have an *elephant's child* profile. In addition, Table 8 shows the percentage of elephant's child agents that the argumentative agent (the initiator) is able to persuade to propose its solution as the best option to solve a ticket. Elephant's child agents have a weird profile that challenges any position proposed by other agents, whatever this position is. This agents are used to overload the system with useless interactions between the agents. An elephant's child agent only accepts the position of other agent if it challenges it and the other agent wins the debate. This has more probability of occurring when agents have less knowledge about the domain and thus, less knowledge to generate positions and arguments. Therefore, whatever strategy the initiator follows, the percentage of agreement overpass only the 50% if agents have low knowledge about the domain and the initiator is able to persuade the 28.57% of agents, as shown in the tables, and decreases as the amount of domain knowledge increases. However, in most cases elephant's child agents are attacking positions that they are able to generate and support and thus, the attacked agents are defeated and withdraw their positions, which prevents the effective development of agreement processes, specially when agents have more domain knowledge and are able to propose more solutions.

Table 9 shows the percentage of times that an agreement about the best solution to apply is reached when the argumentative agent is arguing with agents that have an *open-minded* profile. Also, Table 10 shows the percentage of open-minded agents that the argumentative agent (the initiator) is able to persuade to propose its solution as the best option to solve a ticket. An open-minded agent only challenges the positions of other agents if it has not been able to generate such positions (they are not its proposed

Strategy / Domain Knowledge	**ST1**	**ST2**	**ST3**	**ST4**	**ST5**	**ST6**
LOW	66.67%	66.67%	66.67%	66.67%	66.67%	66.67%
MEDIUM	16.67%	16.67%	16.67%	16.67%	16.67%	16.67%
HIGH	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Table 7: Agreement Percentage with Elephant's Child agents.

Strategy / Domain Knowledge	**ST1**	**ST2**	**ST3**	**ST4**	**ST5**	**ST6**
LOW	28.57%	28.57%	28.57%	28.57%	28.57%	28.57%
MEDIUM	15.48%	15.48%	15.48%	15.48%	15.48%	15.48%
HIGH	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Table 8: Percentage of Elephant's Child agents persuaded.

position or a position in its list of potential positions to propose). However, what specially distinguishes the behaviour of this agent from other agent profiles is the fact that it accepts the position of an agent that has started and won a dialogue with it, although this position is not in its list of potential positions.

As for agreeable and disagreeable agents, if the knowledge that the agents have about the domain is low, no agreement is reached. For the case of having a medium amount of domain knowledge, the agreement is reached easily if the initiator follows a strategy that selects the most potentially persuasive arguments or those that cover as much justification elements as possible (ST1 and ST6). If the initiator is able to better defend its position with more persuasive arguments or more support elements, it ensures that its position will prevail accepted and thus, the defeated open-minded agents will withdraw their positions and agree to propose the initiator's as the best solution to apply. Thus, Table 10 shows a higher percentage of agents persuaded if the initiator follows ST1 and ST6. However, if agents have high knowledge about the domain, the percentage of agreements and agents persuaded are clearly higher when the initiator follows ST2 and ST6. This demonstrates that when the initiator agent has more knowledge and hence, more support degree (higher probability of being accepted at the end of the dialogue) and more elements to justify its positions and arguments, open-minded agents easily accept the position of the initiator and withdraw theirs when they attack the initiator's position.

Strategy / Domain Knowledge	**ST1**	**ST2**	**ST3**	**ST4**	**ST5**	**ST6**
LOW	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
MEDIUM	*41.67%*	33.33%	25.00%	33.33%	16.67%	*41.67%*
HIGH	47.78%	61.11%	47.78%	52.22%	50.00%	*62.78%*

Table 9: Agreement Percentage with Open-Minded agents.

Table 11 shows the percentage of times that an agreement about the best solution to apply is reached when the argumentative agent is arguing with agents that also have an *argumentative* profile. Also, Table 12 shows the percentage of agents with its same profile that an argumentative agent (the initiator) is able to persuade to propose its solution as the best option to solve a ticket. An argumentative agent only challenges positions of other agents when they have proposed a position different from its position. In addition, this agent profile accepts such positions that it attacks when the opponent wins the confrontation.

Again, as in most strategy tests, low knowledge about the domain results in simple dialogues that do not reach any agreement and the solution proposed is not agreed by all agents. For a medium amount of knowledge about the domain, the percentage of agreements reached is higher when the initiator follows a

Domain Knowledge \ Strategy	**ST1**	**ST2**	**ST3**	**ST4**	**ST5**	**ST6**
LOW	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
MEDIUM	*28.57%*	23.81%	16.67%	23.81%	11.90%	27.38%
HIGH	37.46%	48.57%	37.86%	44.21%	38.89%	*51.19%*

Table 10: Percentage of Open-Minded agents persuaded.

strategy that minimises the probability that their positions and arguments are attacked (ST3). In fact, if positions are not attacked at all, they prevail as potential candidates to be selected as the final solution for the ticket requested. Note that in this test, all agents accept positions and arguments of other agents under the same circumstances, so decreasing the probability of a position to be attacked, increases its probability of being accepted at the end of the dialogue.

If argumentative agents have a high knowledge about the domain, many of them are able to propose accurate solutions and defend them against attacks. Therefore, such strategies that allow the initiator to better defend its positions and arguments by increasing the probability of being accepted (ST2) or by reducing the number of potential attacks get better agreement percentages (ST4). If we understand that more acceptable and less attackable arguments lead to shorter dialogues, the good results achieved when the initiator follows a strategy that selects those positions and arguments that produced shorter dialogues in the past (ST5) can be viewed as a logical consequence of the good performance of ST2 and ST4. Table 12 shows a slightly higher percentage of persuaded agents when the initiator has deep knowledge about the domain and the number of potential attacks is decreased. As less attacks are received, less effort the initiator needs to convince other agents that attack its position to agree and accept it as the best solution to apply for the requested ticket. Nevertheless, all ST2, ST4 and ST5 strategies get good percentages of persuaded agents for this amount of domain knowledge.

Domain Knowledge \ Strategy	**ST1**	**ST2**	**ST3**	**ST4**	**ST5**	**ST6**
LOW	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
MEDIUM	38.89%	47.22%	*50.00%*	33.33%	33.33%	41.67%
HIGH	82.22%	*100.00%*	82.22%	*100.00%*	*100.00%*	86.11%

Table 11: Agreement Percentage with Argumentative agents.

Domain Knowledge \ Strategy	**ST1**	**ST2**	**ST3**	**ST4**	**ST5**	**ST6**
LOW	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
MEDIUM	22.22%	29.76%	*30.95%*	21.43%	23.81%	29.76%
HIGH	68.57%	80.56%	66.98%	*81.35%*	80.56%	71.43%

Table 12: Percentage of Argumentative agents persuaded.

Summarizing, Tables 13 and 14 show, for each strategy, the agent profiles that have achieved in average (for all amounts of domain knowledge) the highest percentage of agreement and the highest percentage of agents that the argumentative agent is able to persuade, respectively. For all strategies, the best results in the number of agreements reached and agents persuaded are obtained when all agents are argumentative. Among them, the strategy that guides the agent to propose positions and arguments that have a higher potential probability of being accepted (ST2) is the winning strategy for our customer support domain, followed closely by the strategy that makes the agent to minimise the number of potential attacks that its positions and arguments can receive (ST4). It seems reasonable that if we are measuring the percentage of agreements reached and the percentage of agents persuaded to reach them, that strategy that increases the acceptance probability of positions and arguments performs better. Also, if agents receive less number of attacks, it can be understood as a consequence of proposing positions and arguments that suit the objectives of more agents.

For the rest of profiles, agreeable agents are more easily persuaded with positions and arguments that have a low probability of being attacked (probably because agreeable agents are also able to generate them); disagreeable agents are more easily persuaded with positions and arguments that will potentially get less number of attacks (probably because they are the same that disagreeable agents have proposed); and open-minded agents are easily persuaded with positions and arguments that have more elements that justify them (and hence the initiator has more elements to rebut attacks and win the dialogue). As pointed out before, elephant's child agents show a weird behaviour that gets low percentages of agreement and agents persuaded no matter which strategy the initiator follows. However, in most cases elephant's child agents are attacking positions that they are able to generate and support and thus, the attacked agents are defeated and withdraw their positions, which prevents the development of an effective agreement process.

Strategy / Agent Profile	ST1	ST2	ST3	ST4	ST5	ST6
Agreeable	38.10%	30.95%	*35.71%*	25.00%	16.67%	19.05%
Disagreeable	29.76%	13.10%	8.33%	*34.52%*	16.67%	17.86%
Elephant's Child	7.14%	7.14%	7.14%	7.14%	7.14%	7.14%
Open-Minded	41.90%	47.62%	38.33%	40.24%	35.71%	*48.33%*
Argumentative	**51.90%**	**66.67%**	**60.24%**	**64.29%**	**57.14%**	**58.33%**

Table 13: Average agreement percentage for all agent profiles.

Strategy / Agent Profile	ST1	ST2	ST3	ST4	ST5	ST6
Agreeable	28.57%	24.49%	*26.36%*	18.71%	13.27%	12.93%
Disagreeable	22.28%	10.88%	6.63%	*24.15%*	13.61%	14.97%
Elephant's Child	6.63%	6.63%	6.63%	6.63%	6.63%	6.63%
Open-Minded	30.85%	35.10%	27.45%	31.70%	25.85%	*36.22%*
Argumentative	**38.91%**	**49.83%**	**44.52%**	**48.13%**	**44.73%**	**45.92%**

Table 14: Average percentage of agents persuaded for all agent profiles.

5 Discussion

As stated in [19], bibliography about dialogue strategies in argumentation is hardly found. The few theoretic research performed to date in the area of argumentation in MAS follows two differentiated approaches: the heuristic and the game-theoretic. However, there is no consensus on the definition of a strategy and on the parameters necessary for its definition [2]. Consequently, there is no standard methodology for the definition of argumentation strategies. A first attempt to model the process of strategy construction for negotiation dialogues was published in [21]. This work proposes a methodology for designing heuristic negotiation strategies that guides the strategy designer along several stages to produce modular specifications of tactics and strategies.

The literature on strategies for argumentation provides different definitions for the notion of strategy. Several examples are:

> "The strategy is a decision problem in which an agent tries to choose among different alternatives the best option, which according to its beliefs, will satisfy at least its most important

goals [2].”

> “Private strategies, as adopted by an individual agent, specify the dialogue move(s) the agent is willing to utter, according to its own objectives and other personal characteristics.[13]”

> “A strategy of an agent specifies a complete plan that describes what action the agent takes for every decision that a player might be called upon to take, for every piece of information that the player might have at each time that it is called upon to act. Thus a strategy for an agent would specify for each possible subset of arguments that could define its type (the set of arguments that the agent is capable of putting forward), what set of arguments to reveal.[17, Chapter 16]”

Therefore, there are different approaches to the study of strategies in argumentation frameworks. On one hand, preliminary works studied the concept of strategy as developing heuristics for move selection in argumentation dialogues. A first contribution was provided in [7]. In this work the author defines a *Toulmin dialogue game machine* and proposes some heuristics for move selection. The acceptability of the arguments is computed by using some Toulmin-like rules. A similar work is the one presented in [4], which proposes heuristics for move selection on the context of persuasion and negotiation dialogues. This research defends a three-level approach of strategy, inspired on naturally occurring dialogues between humans. The levels identified are:

- maintaining the focus of the dispute.
- building one’s point of view or destroying the opponent’s one.
- selecting the method to fulfil the objective set at levels 1 and 2.

While levels 1 and 2 refer to *strategy* (planning the line of argumentation), level 3 refers to *tactic* (the mean to reach the aims fixed at the strategical level). Then, the account for strategy proposed follows three steps to develop strategies:

- define some agent profile: agreeable (accept whenever possible), disagreeable (only accept when there is no reason not to), open-minded (only challenge when necessary), argumentative (challenge whenever possible), or elephant child (question whenever possible).
- choose to build or destroy.
- choose some appropriate argumentative content.

Opposite to the former work, this work computes argument acceptability by using a more general Dung’s-like argumentation framework. In a subsequent work, the author studied the notion of strategy for selecting offers during a negotiation dialogue [3], proposing different agent’s profiles and different criteria for the notions of acceptability and satisfiability of offers. Also, [2] views argumentation strategies as a two steps decision process: i) to select the type of act to utter at a given step of a dialogue, and ii) to select the content which will accompany the act. Thus, an agent tries to choose among different alternatives the best option, which according to its beliefs, will satisfy at least its most important goals. There are two types of goals: *strategic goals*, which help an agent, on the basis of the strategic beliefs, to select the type of act to utter; and *functional goals*, which help an agent to select, on the basis of the basic beliefs, the content of a move. Then, the work proposes a formal model for defining strategies. The model takes as input the strategic and the functional goals together with the strategic and basic beliefs and returns the next move (act plus its content) to play. Then, the model assesses each alternative by constructing the set of supporting arguments for each one and evaluating their quality.

The agent profiles of [4] were also considered in [13] to develop different types of strategies. This work proposes an argument-based framework for representing communication theories of agents that can take into account the conformance to society protocols, private tactics of individual agents, strategies that reflect different types of personal attitudes (agents’ profiles) and adaptability to the particular external circumstances at the time when the communication takes place. Although the authors do not provide a clear structure and definition for their notion of agent society, social relations between agents are captured in the form of preference rules that affect the tactic component of an agent and help it to decide the next move in a dialogue.

On the other hand, a different approach follows a game-theoretic view to the study of argumentation strategies. This is the case of the work proposed in [23], where the probability of a conclusion is calculated using a standard variant of defeasible logic, in combination with standard probability calculus. In this approach the exchange of arguments is analysed with game-theoretic tools, yielding a prescriptive account of the actual course of play. Other game-theoretic approach for the study or argumentation strategies in negotiation dialogues was presented in [17]. This approach uses the paradigm of *Argumentation Mechanism Design (ArgMD)* for designing and analysing argument evaluation criteria among self-interested agents using game-theoretic techniques. Mechanism design (MD) is a sub-field of game theory concerned with determining the game rules that guarantee a desirable social outcome when each self-interested agent selects the best strategy for itself.

The approach analyses strategy-proofness under grounded semantics for a specific type of arguments, the so-called *focal arguments* (the arguments that agents are specially interested in being accepted). In a preliminary work [16], the authors restricted the analysis to the case where agents use a specific type of preference criteria, the *individual acceptability maximising preference* criteria. Following this criteria, every agent attempts to maximise the number of its arguments that are accepted. In further research [18] the ArgMD approach has been applied to more realistic situations in which each agent has a single focal argument it wishes to have accepted. The authors demonstrate for both preference criteria that if each agent's type (characterised as the set of argument that an agent can bring up) corresponds to a conflict-free set of arguments which does not include (in)direct defeats, the *grounded direct argumentation mechanism* for this argumentation framework is strategy-proof. Opposite to the heuristic-based approaches, the goal of this game-theoretic approach is to design rules that ensure, under precise conditions, that agents have no incentive to manipulate the outcome of the game by hiding arguments or lying (how to ensure the truth in an argumentation framework).

As pointed out in Section 3, we are closer to the heuristic approach of [2] rather than to the game theoretic ones. One one hand, game theoretic approaches are usually applied to abstract argumentation frameworks where the strategies of agents determine which argument(s) they will reveal in each argumentation step. Common objectives of these works are to study the conditions under which the outcome of the game is not affected by the strategic decisions of the agents or to predict the outcome of the game. In contrast, we define dialogue strategies on basis of the knowledge that agents have about the domain, previous argumentation experiences and the social context (the roles, preferences over values and dependency relations among agents and groups). In doing so, we take into account the specific structure of arguments and the knowledge resources of our framework and design strategies to help agents to take advantage over other participants in the argumentation dialogue.

On the other hand, in our application domain there is not a pre-defined utility function about the payoff that an agent gets for the fact of winning the dialogue or having accepted more or less arguments, which is one of the common assumptions in game theoretic approaches for strategic argumentation. Finally, game theory assumes complete knowledge of the space of arguments proposed in the argumentation framework. This assumption is unrealistic in an argumentation dialogue between heterogeneous agents which have individual and private knowledge resources to generate arguments.

Further disadvantages of applying game theory to devise dialogue strategies in argumentation frameworks are similar as those reported in [12] for the problem of applying game theory to automated negotiation. First, game theoretic studies of rational choice in multi-agent encounters typically assume that agents are allowed to select the best strategy from the space of all possible strategies, by considering all possible interactions. This is computationally intractable when the space of possible choices growths. Also, game theory assumes that it is possible to characterise an agent's preferences about possible outcomes. This is hard to define for agents that represent humans that are engaged in an argumentation process. Our alternative solution is to define preferences over values instead of preferences over dialogue outcomes and use them to guide the agents' choices.

From our point of view, an interesting role that the CBR methodology can play in argumentation processes in MAS is to generate heuristic argumentation strategies based on previously acquired experience. Note that one of the main advantages of using CBR to manage argumentation in MAS is that it allows agents to learn from the process. The few case-based argumentation frameworks found in the literature partially store the information about the current argumentation dialogue in the form of cases when the process finishes [10]. In addition, the agents of the *AMAL* case-based argumentation framework presented in [14] can also learn during the argumentation dialogue by storing in their case-bases the cases that they receive from other agents. However, by contrast with our proposal, they do not learn how to predict the behaviour of particular agents or the expected development of the dialogue, but only increase their own knowledge with the knowledge that other agents share with them.

As shown in this report, CBR can also be used to learn the behaviour of specific agents and generate arguments to perform a strategic argumentation that would easily persuade different types of agents. Some preliminary steps in this way have already been taken in related work. The fist attempt to use CBR to provide information for building agents' profiles was performed in the *PERSUADER* system [26]. In this framework a mediator agent uses the information about previous negotiation processes stored in the case-base to develop the behavioural model of an unknown agent and devise the best way to persuade it. Similarly, in the negotiation framework proposed in [25] the information of the cases is used to decide which type of arguments are best suited to convince an agent with a specific profile and to infer other parameters that influence the negotiation process (e.g. time constraints, resources usage, etc.). Nevertheless, in both cases the argumentation strategy is highly domain dependent and completely relies on previous knowledge. Although in PERSUADER the agents' models can be dynamically updated, the preference order that determines which argument must be ultimately posed depends on a pre-established hierarchy.

In a more dynamic and online way, the case-base could be used to store information about the agents' profile that could be gathered either by observing the current agents' behaviour, by learning information that the agents send during the dialogue or as a result of inquiry and information seeking processes. Therefore, this information could be used in the current argumentation process to generate and select better arguments to put forward and to evaluate the incoming ones. Case-based argumentation strategies have to be further investigated and there is still much work to do in this area.

6 Conclusions

In this report we have presented different argumentation strategies (as combinations of an agent profile and different tactics) that an agent can follow to argue with other agents in a case-based argumentation framework for agent societies. To evaluate this proposal, the framework has been implemented in a real customer support application.

The tests performed evaluate the percentage of agreements reached and the percentage of agents of different profiles persuaded by an argumentative agent. The results obtained demonstrate that if all participants of the argumentation process have an argumentative profile, more agreements are reached and more agents support the final solutions provided by the system. For all strategies, the best results are obtained when all agents are argumentative. Among them, the strategies that guide the agent that follows them to propose positions and arguments that have a higher potential probability of being accepted (ST2) and that minimise the number of potential attacks that an agent can receive (ST4) are the winning strategies. Future work will evaluate further combinations of profile-tactic for the initiator and also for the rest of the agents participating in the dialogue.

We have assumed in this example that agents do their best to win the argumentation dialogue, thus following a persuasion dialogue, since in this way they can get economical rewards and increase prestige. Despite that, those solutions that are better supported prevail. Hence, if agents do not follow a dialogue strategy that deviates the final outcome of the dialogue to fit their individual objectives, no matter if they give rise to the best solution to apply, the system reaches agreements that produce high quality solutions for the tickets received. This assumption has allowed us to perform more comprehensive tests with the small amount of data that we have and to check the advantages of following different dialogue strategies and of the amount of available knowledge about the preferences of other agents. However, a cooperative approach where agents are not interested and collaborate to reach the best agreement would be appropriate for this example and will be implemented and evaluated in the future.

Funding

This work is supported by the Spanish government grants [CONSOLIDER-INGENIO 2010 CSD2007-00022, TIN2008-04446, and TIN2009-13839-C03-01] and by the GVA project [PROMETEO 2008/051].

References

[1] A. Aamodt and E. Plaza. Case-based reasoning: foundational issues, methodological variations and system approaches. *AI Communications*, 7(1):39–59, 1994.

[2] L. Amgoud and N. Hameurlain. A formal model for designing dialogue strategies. In *5th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS-06*, pages 414–416. ACM Press, 2006.

[3] L. Amgoud and S. Kaci. On the study of negotiation strategies. In *AAMAS 2005 Workshop on Agent Communication, AC-05*, pages 3–16. ACM Press, 2005.

[4] L. Amgoud and N. Maudet. Strategical considerations for argumentative agents (preliminary report). In *9th International Workshop on Non-Monotonic Reasoning, NMR-02*, LNAI, pages 399–407. Springer, 2002.

[5] L. Amgoud and S. Parsons. Agent dialogues with conflicting preferences. In *5th International Workshop on Agent Theories, Architectures and Languages, ATAL-01*, LNAI, pages 1–17. Springer, 2001.

[6] T. Bench-Capon and G. Sartor. A model of legal reasoning with cases incorporating theories and values. *Artificial Intelligence*, 150(1-2):97–143, 2003.

[7] T. J. Bench-Capon. Specification and implementation of toulmin dialogue game. In *International Conferences on Legal Knowledge and Information Systems, JURIX-98*, Frontiers of Artificial Intelligence and Applications, pages 5–20. IOS Press, 1998.

[8] F. Dignum and H. Weigand. Communication and Deontic Logic. In R. Wieringa and R. Feenstra, editors, *Information Systems - Correctness and Reusability. Selected papers from the IS-CORE Workshop*, pages 242–260. World Scientific Publishing Co., 1995.

[9] S. Heras. *A Case-Based Argumentation Framework for Agent Societies*. PhD thesis, Departamento de Sistemas Informáticos y Computación. Universitat Politècnica de València, 2011.

[10] S. Heras, V. Botti, and V. Julián. Challenges for a CBR framework for argumentation in open MAS. *Knowledge Engineering Review*, 24(4):327–352, 2009.

[11] S. Heras, J. A. García-Pardo, R. Ramos-Garijo, A. Palomares, V. Botti, M. Rebollo, and V. Julián. Multi-domain case-based module for customer support. *Expert Systems with Applications*, 36(3):6866–6873, 2009.

[12] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated negotiation: prospects, methods and challenges. *International Journal of Group Decision and Negotiation*, 10(2):199–215, 2001.

[13] A. Kakas, N. Maudet, and P. Moraitis. Modular Representation of Agent Interaction Rules through Argumentation. *Autonomous Agents and Multi-Agent Systems*, 11:189–206, 2005.

[14] S. Ontañón and E. Plaza. Learning and joint deliberation through argumentation in multi-agent systems. In *7th International Conference on Agents and Multi-Agent Systems, AAMAS-07*. ACM Press, 2007.

[15] I. Rahwan. Argumentation in multi-agent systems. *Autonomous Agents and Multiagent Systems, Guest Editorial*, 11(2):115–125, 2006.

[16] I. Rahwan and K. Larson. Mechanism design for abstract argumentation. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, volume 2, pages 1031–1038. ACM Press, 2008.

[17] I. Rahwan and K. Larson. Argumentation and Game Theory. *Argumentation in Artificial Intelligence*, pages 321–339, 2009.

[18] I. Rahwan, K. Larson, and F. Tohmé. A characterisation of strategy-proofness for grounded argumentation semantics. In *21st International Jont Conference on Artifical Intelligence, IJCAI-09*, pages 251–256. Morgan Kaufmann Publishers Inc., 2009.

[19] I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons, and L. Sonenberg. Argumentation-based negotiation. *The Knowledge Engineering Review*, 18(4):343–375, 2003.

[20] I. Rahwan and G. Simari, editors. *Argumentation in Artificial Intelligence*. Springer, 2009.

[21] I. Rahwan, L. Sonenberg, N. R. Jennings, and P. McBurney. STRATUM: A methodology for designing heuristic agent negotiation strategies. *Applied Artificial Intelligence*, 21(6):489–527, 2007.

[22] E. L. Rissland, K. D. Ashley, and L. K. Branting. Case-based reasoning and law. *The Knowledge Engineering Review*, 20(3):293–298, 2006.

[23] B. Roth and A. Rotolo. Strategic argumentation: a game theoretical investigation. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Law*, pages 81–90. ACM Press, 2007.

[24] C. Sierra, V. Botti, and S. Ossowski. Agreement Computing. *KI - Künstliche Intelligenz*, DOI: 10.1007/s13218-010-0070-y, 2011.

[25] L.-K. Soh and C. Tsatsoulis. A real-time negotiation model and a multi-agent sensor network implementation. *Autonomous Agents and Multi-Agent Systems*, 11(3):215–271, 2005.

[26] K. Sycara. Persuasive argumentation in negotiation. *Theory and Decision*, 28:203–242, 1990.

[27] D. Walton, C. Reed, and F. Macagno. *Argumentation Schemes*. Cambridge University Press, 2008.