Dispute Resolution Using Argumentation-Based Mediation

Tomas Trescak[1], Carles Sierra[2], Simeon Simoff[1], and
Ramon Lopez de Mantaras[2]

[1] School of Computing, Engineering and Mathematics, University of Western Sydney, Australia
t.trescak@uws.edu.au, s.simoff@uws.edu.au
[2] Artificial Intelligence Research Institute, CSIC, Barcelona, Spain
sierra@iiia.csic.es, mantaras@iiia.csic.es

Abstract. Mediation is a process, in which both parties agree to resolve their dispute by negotiating over alternative solutions presented by a mediator. In order to construct such solutions, mediation brings more information and knowledge, and, if possible, resources to the negotiation table. The contribution of this paper is the automated mediation machinery which does that. It presents an argumentation-based mediation approach that extends the logic-based approach to argumentation-based negotiation involving BDI agents. The paper describes the mediation algorithm. For comparison it illustrates the method with a case study used in an earlier work. It demonstrates how the computational mediator can deal with realistic situations in which the negotiating agents would otherwise fail due to lack of knowledge and/or resources.

1 Introduction and Motivation

Dispute resolution is a complex process, depending on the will of involved parties to reach consensus, when they are satisfied with the result of negotiation, which allows them to partially or completely fulfil their goals with the available resources. In many cases, such negotiation depends on searching for alternative solutions, which requires an extensive knowledge about the disputed matter for sound argumentation. Such information may not be available to the negotiating parties and negotiation fails. Mediation, a less radical alternative to arbitration, can assist both parties to come to a mutual agreement. This paper presents an argumentation-based mediation system that builds on previous works in the field of argumentation-based negotiation. It is an extension of the work presented in [1] and focuses on problems where negotiation stalled and had no solution. In [1] agents contain all the knowledge and resources needed to resolve their dispute - a relatively strong assumption in the context of real world negotiations. Agents present arguments, which their opponent can either accept, reject, or they can negotiate on a possible solution. As mentioned earlier, lacking knowledge or resources may lead to an unsuccessful negotiation. In many cases, such knowledge or even alternative resources may be available, but agents are not aware of them.

Our extension proposes a role of a trust-worthy mediator that possesses extensive knowledge about possible solutions of mediation cases, which it can adapt to the current case. *Mediator also has access to various resources that may help to resolve the dispute.* Using this knowledge and resources, as well as knowledge and resources obtained from agents, the mediator creates alternative solutions, which become subject to further negotiation. Furthermore, mediator is guaranteed to be neutral and considered trust-worthy by all interested parties.

In the next section, we summarise related work in the field of automatic mediation and argumentation-based negotiation. In Section 3, we recall the agent architecture proposed by Parsons et al. [1] and extend it with the notion of resources for the purposes of the mediation system. Section 4 presents our mediation algorithm. In Section 5, we revisit the home improvement agents example from [1] and apply our mediation process. Section 6 concludes this work.

2 Previous Work

Computational mediation has recognized the role of the mediator as a problem solver. The `MEDIATOR` [2] focused on case-based reasoning as a single-step for finding a solution to a dispute resolution problem. The mediation process was reduced to a one-step case-based inference, aimed at selecting an abstract "mediation plan". The work did not consider the value of the actual dialog with the mediated parties. The `PERSUADER` [3] deployed mechanisms for problem restructuring that operated over the goals and the relationships between the goals within the game theory paradigm, applied to labor management disputes. To some extent this work is a precursor of another game-theoretic approach to mediation, presented in [4] and the interest-based negotiation approach in [5]. Notable are recent game-theoretic computational mediators `AutoMed` [6] and `AniMed` [7] for multi-issue bilateral negotiation under time constraints. They operate within known solution space, offering either specific complete solutions (`AutoMed`) or incremental partial solutions (`AniMed`). Similar to the mediator proposed in the 'curious negotiator' [8], both mediators monitor negotiations and intervene when there is a conflict between negotiators. The `Family_Winner` [9] manipulative mediator aimed at modifying the initial preferences of the parties in order to converge to a feasible and mutually acceptable solution. This line of works incorporated "fairness" in the mediation strategies [10].

In real settings information only about negotiation issues is not sufficient to derive the outcome preferences [11]. An exploratory study [12] of a multiple (three) issue negotiation setting suggests the need for developing integrative (rather than position-based) negotiation processes which take into account information about the motivational orientation of negotiating parties. Incorporation of information beyond negotiation issues has been the focus of a series of works related to information-based agency [13, 14, 15]. Value-based argumentation frameworks [16], interest-based negotiation [5] and interest-based reasoning [11] considers the treatment of any kind of motivational information that leads to a preference in negotiation and decision making.

In this paper we propose a new mechanism for automatic mediation using *argumentation-based negotiation* (ABN) as a principal framework for mediation. ABN systems evolved from classical argumentation systems, bringing power to agents to resolve potential dispute deadlocks by persuasion of agents in their beliefs and finding common acceptance grounds by negotiation [17, 1, 18, 19]. ABN is performed by exchanging arguments, which represent a stance of an agent related to the negotiated subject and constructed from beliefs of the agent. Such a stance can support another argument of the agent, explain why a given offer is rejected, or provide conditions upon which the offer would be accepted. Disputing parties can modify their offer or present a counter-offer, based on the information extracted from the argument. Arguments can be used to attack [20] other arguments, supporting or justifying the original offer. With certain level of trust between negotiating agents, arguments serve as knowledge exchange carriers [1] - here we use such mechanisms to exchange information between negotiating parties and the mediator. The decision of whether to trust the negotiating party or not is a part of the *strategy* of an agent. Different strategies are proposed in [21, 22, 23]. Apart from the strategy, essential are the *reasoning mechanisms* and *negotiation protocols*. Relevant to this work are logic frameworks that use argumentation as the key mechanism for reasoning [24, 25, 26, 27]. *Negotiation protocols*, which specify the negotiation procedures include either finite-state machines [1], or functions based on the previously executed actions [28]. The reader is referred to [29] for the recent state-of-the-art in ABN frameworks.

Our ABN framework for mediation allows us to seamlessly design and execute realistic mediation process, which utilises the power of argumentation, using agent logics and a negotiation procedure to search for the common agreement space. We have decided to extend the ABN framework in [1], due to the clarity of its logics. In the next section we recall the necessary aspects of the work in [1]. We describe the agent architecture in the ABN systems and define the components that we reuse in our work. Our agents *reason* using argumentation, based on a domain dependent theory specified in a first-order logic. Within the theory, we encode *agent strategies*, by defining their planning steps. Apart from agent theories, *strategy* is defined also in *bridge rules*, explained further in the text. We do not explore a custom protocol, therefore we adopt the one from [1].

3 Agent Architecture

The ABN system presented in [1] is concerned with BDI agents in a multi-context framework, which allows distinct theoretical components to be defined, interrelated and easily transformed to executable components. The authors use different contexts to represent different components of an agent architecture, and specify the interactions between them by means of the bridge rules between contexts. We recall briefly the components of the agent architecture within the ABN system in [1] and add a new "resources" component for mediation purposes.

Units are structural entities representing the main components of the architecture. There are four units within a multi-context BDI agent, namely: the **C**ommunication unit, and units for each of the **B**eliefs, **D**esires and **I**ntentions.

Bridge rules connect units, which specify internal agent architecture by determining their relationship. Three well-established sets of relationships for BDI agents have been identified in [30]: *strong realism, realism and weak realism.* In this work, we consider strongly realist agents.

Logics is represented by declarative languages, each with a set of axioms and a number of rules of inference. Each unit has a single logic associated with it. For each of the mentioned B, D, I, C units, we use classical first-order logic, with special predicates B, D and I related to their units. These predicates allow to omit the temporal logic CTL modalities as proposed in [30].

Theories are sets of formulae written in the logic associated with a unit. For each of the four units, we provide domain dependent information, specified as logical expressions in the language of each unit.

Bridge rules are rules of inference which relate formulae in different units. Following are bridge rules for strongly realist BDI agents:

$$\begin{array}{rr} I : I(\alpha) \Rightarrow D : D(\lceil\alpha\rceil) & B : \neg B(\alpha) \Rightarrow D : \neg D(\lceil\alpha\rceil) \\ D : \neg D(\alpha) \Rightarrow I : \neg I(\lceil\alpha\rceil) & C : done(e) \Rightarrow B : B(\lceil done(e)\rceil) \\ D : D(\alpha) \Rightarrow B : B(\lceil\alpha\rceil) & I : I(\lceil does(e)\rceil) \Rightarrow C : does(e) \end{array}$$

Resources are our extension of the contextual architecture of strongly realist BDI agents. Each agent can possess a set of resources R^v with a specific importance value for its owner. This value may determine the order in which agents are willing to give up their resources during the mediation process. We define a value function $v : \$ \rightarrow \mathbb{R}$, which for each resource ϕ specifies a value $\vartheta \in< 0, 1 >$, $v(\phi) = \vartheta$. Set R^v is ordered according to function v.

Units, *logics* and *bridge rules* are static components of the mediation system. All participants have to agree on them before the mediation process starts. *Theories* and *resources* are dynamic components, they change during the mediation process depending on the current state of negotiation.

4 Mediation Algorithm

In a mediation process both parties try to resolve their dispute by negotiating over alternative solutions presented by a mediator. Such solutions are constructed, using available knowledge and resources. Agent knowledge is considered private and is not shared with the other negotiating party. Resources to obtain alternative solutions may have a high value for their owners or be entirely missing. Thus, we propose that the role of the mediator is to obtain enough knowledge and resources to be able to construct a new solution. The mediator presents a possible solution to agents (in the form of an *argument*), which they either approve, or reject (*attack*). For the purposes of this paper we follow Dong'e notion of attack [20]. Parties can negotiate over a possible solution to come to a mutual agreement. Below we formally define the foundations of our algorithm.

Definition 1. *Δ is a set of formulae in language L. An argument is a pair (Φ, ω), $\Phi \subseteq \Delta$ and $\omega \in \Delta$ such that: (1) $\Phi \nvdash \perp$; (2) $\Phi \vdash \omega$; and (3) Φ is a minimal subset of Δ satisfying 2.*

A *mediation game* is executed in one or more rounds, during which both mediator and agents perform various actions in order to resolve the dispute. Algorithm 1 describes our proposal of the mediation game. In the beginning of each round, agents α and β have an opportunity to present new knowledge to the mediator μ. This new knowledge is helping their case, or helping to resolve the dispute. Agents can either present knowledge in the form of formulas from their theory or new resources. Resources can be presented in ascending order of importance, one resource in each round or altogether, depending on the strategy of agents. The mediator obtains knowledge by executing function $\Gamma_i^\mu \leftarrow GetKnowledge(i)$, where $i \in \{\alpha, \beta\}$. The mediator incorporates knowledge Γ_i^μ into theory Γ_μ, obtaining Γ'_μ. Please note, that the belief revision operator $\oplus$ is responsible for automatic elimination of conflicting beliefs from the theory. Belief revision operator uses argumentation techniques to find the minimal set of non-conflicting arguments. Using the knowledge in Γ'_μ, the mediator tries to construct a new *solution* by executing the $CreateSolution(\Gamma'_\mu)$ function. If the *solution* does not exist and agents did not present new knowledge in this round, mediation fails. Therefore, it is of utmost importance that agents try to introduce knowledge in each round. In the next step, the possible outcomes are:

- When both agents accept the *solution*, mediation finishes with success.
- When both agents reject the *solution*, mediator adds the incorrect solution $\neg solution$ and the explanation of the rejection $\Gamma_i^{\mu'}$ from both agents to its knowledge Γ'_μ and starts a new mediation round.
- When only one agent rejects the solution, a new negotiation process is initiated, where agents try to come to a mutual agreement (e.g. partial division of a specific item) resulting to $solution'$. If this negotiation is successful, the mediator records $solution'$ as a new solution and finishes mediation with a success. If the negotiation fails, the mediator adds the explanation of the failure $\Gamma_i^{\mu''}$ and the failed *solution* to Γ'_μ and starts a new mediation round.

The mediation process continues till a resolution is obtained, or fails, when no new solution can be obtained, and no new knowledge can be presented. In the next section, we revisit the example of home improvement agents from [1] and apply the mediation algorithm.

5 Case Study: Revisiting Home Improvement Agents

In this example, agent α is trying to hang a picture on the wall. Agent α knows that to hang a picture it needs a nail and a hammer. Agent α only has a screw, and a hammer, but it knows that agent β owns a nail. Agent β is trying to hang a mirror on the wall. β knows that it needs a nail and a hammer to hang the mirror, but β currently possesses only a nail, and also knows that α has a hammer. Mediator μ owns a screwdriver and knows that a mirror can be hung using a screw and a screwdriver.

The difference with the example in [1] is that mediator owns the knowledge and resource needed to resolve the dispute μ and not the agents. This reflects

Input : Agents α, β and the mediator μ. Γ_α, Γ_β and Γ_μ denote the knowledge of α, β and μ, while Γ_α^μ and Γ_β^μ denote the knowledge presented to the mediator μ respectively by α and β. $\oplus$ is a *belief revision operator*
Output: Resolution of the dispute, or $\perp$ if solution does not exists.

repeat
  $\Gamma_\alpha^\mu \leftarrow$ `GetKnowledge` (α); `// Theory and resources from` α
  $\Gamma_\beta^\mu \leftarrow$ `GetKnowledge` (β); `// Theory and resources from` β
  $\Gamma'_\mu \leftarrow \Gamma_\mu \oplus (\Gamma_\alpha^\mu \cup \Gamma_\beta^\mu)$;
  solution $\leftarrow$ `CreateSolution` (Γ'_μ);
  if solution $= \perp$ *and* $\Gamma_\mu = \Gamma'_\mu$ **then**
    return $\perp$; `// Missing new knowledge and no solution`
  end
  if solution $\neq \perp$ **then**
    $\langle result_\alpha, \Gamma_\alpha^{\mu'} \rangle \leftarrow$ `Propose`$(\mu, \alpha,$ solution$)$
    $\langle result_\beta, \Gamma_\beta^{\mu'} \rangle \leftarrow$ `Propose`$(\mu, \beta,$ solution$)$
    $\Gamma'_\mu \leftarrow \Gamma'_\mu \oplus (\Gamma_\alpha^{\mu'} \cup \Gamma_\beta^{\mu'})$
    if $\neg result_\alpha$ *and* $\neg result_\beta$ **then**
      $\Gamma'_\mu \leftarrow \Gamma'_\mu \oplus \neg$solution;
      solution $\leftarrow \perp$;
    else if $\neg result_\alpha$ *or* $\neg result_\beta$ **then**
      $\langle$solution$'$, $\Gamma_\alpha^{\mu''}$, $\Gamma_\beta^{\mu''} \rangle \leftarrow$ `Negotiate` (solution, α, β);
      $\Gamma'_\mu \leftarrow \Gamma'_\mu \oplus (\Gamma_\alpha^{\mu''} \cup \Gamma_\beta^{\mu''})$
      if $\neg$solution$'$ **then**
        $\Gamma'_\mu \leftarrow \Gamma'_\mu \oplus (\neg$solution $\cup \neg$solution$')$
        solution $\leftarrow \perp$;
      else
        solution $\leftarrow$ solution$'$
      end
    end
  end
  $\Gamma_\mu \leftarrow \Gamma'_\mu$;
until solution $\neq \varnothing$;
return solution

Algorithm 1: Mediation algorithm

the reality, when clients seek advice of an expert to resolve their problem. As mentioned in the Section 3, agents α and β are strongly realist BDI agents using related bridge rules and predicate logic. We now define all the dynamic parts of the mediation system, i.e. domain specific agent theory and bridge rules[1].

[1] We adopt following notation: A.* is the theory introduced by the agent α, B.* is the theory of the agent β, M.* is the mediator's theory, G.* is the general theory and R.* are bridge rules

5.1 Agent Theories

What follows, is the private theory Γ_α of the agent α, whose intention is to hang a picture:

$$\text{I} \;:\; I_\alpha(Can(\alpha, hang\_picture)) \tag{A.1}$$
$$\text{B} \;:\; B_\alpha(Have(\alpha, picture)) \tag{A.2}$$
$$\text{B} \;:\; B_\alpha(Have(\alpha, screw)) \tag{A.3}$$
$$\text{B} \;:\; B_\alpha(Have(\alpha, hammer)) \tag{A.4}$$
$$\text{B} \;:\; B_\alpha(Have(\beta, nail)) \tag{A.5}$$
$$\text{B} \;:\; B_\alpha(Have(X, hammer) \wedge Have(X, nail) \wedge Have(X, picture) \rightarrow Can(X, hangPicture)) \tag{A.6}$$

Please note, that agent α, contrarily to the example in [1], no longer knows that a mirror can be hung with a screw and a screwdriver. What follows, is the private theory Γ_β of agent β, whose intention is to hang a mirror.

$$\text{I} \;:\; I_\beta(Can(\beta, hangMirror)) \tag{B.1}$$
$$\text{B} \;:\; B_\beta(Have(\beta, mirror)) \tag{B.2}$$
$$\text{B} \;:\; B_\beta(Have(\beta, nail)) \tag{B.3}$$
$$\text{B} \;:\; B_\beta(Have(X, hammer) \wedge Have(X, nail) \wedge Have(X, mirror) \rightarrow Can(X, hangMirror)) \tag{B.4}$$

Following is the theory Γ_μ of the mediator μ, related to the home improvement agents case (please note, that mediator's knowledge can consist of many other beliefs, for example learned from other mediation cases):

$$\text{B} \;:\; B_\mu(Have(\mu, screwdriver)) \tag{M.1}$$
$$\text{B} \;:\; B_\mu(Have(X, screw) \wedge Have(X, screwdriver) \wedge Have(X, mirror) \rightarrow Can(X, hang\_mirror)). \tag{M.2}$$
$$\text{B} \;:\; B_\mu(Have(X, hammer) \wedge Have(X, nail) \wedge Have(X, mirror) \rightarrow Can(X, hangMirror)) \tag{M.3}$$

We adopt the following theories from [1] with actions that integrate different models reflecting real world processes such as change of ownership, and processes that model decisions and planning of actions. In what follows $i \in \{\alpha, \beta\}$).

Ownership. When an agent (X) gives up artifact (Z) to (Y), (Y) becomes its new owner:

$$\text{B} \;:\; B_i(Have(X, Z) \wedge Give(X, Y, Z) \rightarrow Have(Y, Z)) \tag{G.1}$$

Reduction. If there is a way to achieve an intention, an agent adopts the intention to achieve its preconditions:

$$\text{B} \;:\; B_i(I_j(Q)) \wedge B_i(P_1 \wedge \ldots \wedge P_k \wedge \ldots \wedge P_n \rightarrow Q) \wedge \neg B_i(R_1 \wedge \ldots \wedge R_m \rightarrow Q) \rightarrow B_i(I_j(P_l)) \tag{G.2}$$

Generosity Mediator μ is willing to give up any resource Q

$$\text{B} \;:\; B_\mu(Have(\mu, Q)) \rightarrow \neg I_\mu(Have(\mu, Q)). \tag{G.3}$$

Unicity. When an agent (X) gives an artifact (Z) away, (X) longer owns it:

$$\text{B} \;:\; B_i(Have(X, Z) \wedge Give(X, Y, Z) \rightarrow \neg Have(X, Z)) \tag{G.4}$$

Benevolence. When agent i does not need (Z) and is asked for it by X, i will give Z up:

$$\text{B} \quad : \quad B_i(Have(i,Z) \wedge \neg I_i(Have,i,Z) \wedge Ask(X, i.Give(i,X,Z)) \rightarrow I_i(Give(i,X,Z))) \tag{G.5}$$

Parsimony. If an agent believes that it does not intend to do something, it does not believe that it will intend to achieve the preconditions (i.e. the means) to achieve it:

$$\text{B} \quad : \quad B_i(\neg I_i(Q)) \wedge B_i(P_1 \wedge \ldots \wedge P_j \wedge \ldots \wedge P_n \rightarrow Q) \rightarrow \neg B_i(I_i(P_j)) \tag{G.6}$$

Unique choice. If there are two ways of achieving an intention, only one is intended. Note that we use ∇ to denote exclusive or.

$$\text{B} \quad : \quad B_i(I_i(Q)) \wedge B_i(P_1 \wedge \ldots \wedge P_j \wedge \ldots \wedge P_n \rightarrow Q) \wedge B_i(R_1 \wedge \ldots \wedge R_n \rightarrow Q) \rightarrow B_i(I_i(P_1 \wedge \ldots \wedge P_n)) \nabla B_i(I_i(R_1 \wedge \ldots \wedge R_n)) \tag{G.7}$$

A theory that contains free variables (e.g. X) is considered the *general theory*, while a theory with bound variables (e.g. α or β) is considered the *case theory*. The mediator stores only the *general theory* for its reuse with future cases. In addition, an agent's theory contains rules of inference, such as modus ponens, modus tollens and particularization.

5.2 Bridge Rules

What follows, is a set of domain dependent bridge rules that link inter-agent communication and the agent's internal states.

Advice. When the mediator μ believes that it knows about possible intention I_X of X it *tells* it to X. Also, when mediator μ knows something (ϕ) that can help to achieve intention φ of agent X, mediator *tells* it to X.

$$B_\mu(I_X(\varphi)) \Rightarrow Tell(\mu, X, B_\mu(I_X(\varphi))) \tag{R.1}$$

$$B_\mu(I_X(\varphi)) \wedge B_\mu(\phi \rightarrow I_X(\varphi)) \Rightarrow Tell(\mu, X, B_\mu(\phi \rightarrow I_X(\varphi))) \tag{R.2}$$

Trust in mediator When an agent (i) is told of a belief of mediator (μ), it accepts that belief:

$$C : Tell(\mu, i, B_\mu(\varphi)) \Rightarrow B : B_i(\varphi). \tag{R.3}$$

Request. When agent (i) needs (Z) from agent (X), it asks for it:

$$I : I_i(Give(X,i,Z)) \Rightarrow C : Ask(i, X, Give(X,i,Z)). \tag{R.4}$$

Accept Request. When agent (i) asks something (Z) from agent (X), and it is not in intention of (X) to have (Z), it is given to i:

$$C : Ask(i, X, Give(X,i,Z)) \wedge \neg I_X(Have(X,Z)) \Rightarrow I_i(Give(X,i,Z)). \tag{R.5}$$

5.3 Resources

In Section 3, we have introduced the notion of importance of resources, which defines the order in which agents are giving up their resources during the mediation process. The picture and the hammer depend on the successful accomplishment of agent's α goal and have an importance value of 1. Agent β owns a mirror and a nail, both with importance 1. All other resources have importance 0.

5.4 Argumentation System

Our automatic mediation system uses the ABN system, proposed in [1], which is based on the one proposed in [24]. The system constructs a series of logical steps (arguments) for and against propositions of interest and as such may be seen as an extension of classical logic. In classical logic, an argument is a sequence of inferences leading to a true conclusion. It is summarized by the schema $\Gamma \vdash (\varphi, G)$, where Γ is the set of formulae available for building arguments, $\vdash$ is a suitable consequence relation, φ is the proposition for which the argument is made, and G indicates the set of formulae used to infer φ, with $G \subseteq \Gamma$.

5.5 Mediation

In this section, we follow Algorithm 1 and explain how we can resolve the home improvement agent dispute using automatic mediation. In comparison to Parsons et al. [1], our agents do not possess all the knowledge and resources to resolve their dispute; thus the classical argumentation fails. The mediation algorithm runs in rounds and finishes with:

1. Success, when both agents accept the solution proposed by the mediator.
2. Failure, when the mediator can not create a new solution and no new knowledge or resources are presented in two consecutive rounds.

The algorithm starts with the mediator gathering information about the dispute from both agents (function *GetKnowledge*). In the first round, agents α and β state their goals, which become part of the mediator's beliefs B_μ:

$$\text{B} \; : \; B_\mu(I_\alpha(Can(\alpha, hang\_picture))) \tag{M.4}$$
$$\text{B} \; : \; B_\mu(I_\beta(Can(\beta, hangMirror))) \tag{M.5}$$

With this new theory, the mediator tries to construct a new solution, and it fails. Therefore, in the next round, agents have to present more knowledge or resources. Failing to do so would lead to failure of the mediation process. To speed things up, we assume that agents presented all the necessary knowledge and resources in this single step, although this process can last several rounds depending on the strategy of an agent. For example, if a "cautious" agent owns more than one resource, it chooses to give up the resource with the lowest importance.

B: $B_\mu(Have(\alpha, picture))$ (M.6)

B: $B_\mu(Have(\alpha, screw))$ (M.7)

B: $B_\mu(Have(\alpha, hammer))$ (M.8)

B: $B_\mu(Have(\beta, nail))$ (M.9)

B: $B_\mu(Have(\beta, mirror))$ (M.10)

With this new information, the mediator is finally able to construct the `solution` to the dispute consisting of three different arguments. With the following two arguments, mediator proposes agent β to hang the mirror using the screw and the screwdriver (M.2), and screw can be obtained from the agent α and the screwdriver obtained from the mediator itself (Please note, that this knowledge is part of the support for the presented arguments). The first argument is: $(I_\beta(Give(\alpha, \beta, screw)), P'_\beta)$, where P'_β is[2]:

$$\{(\text{M.2}),(\text{M.5}),(\text{G.2})\} \vdash_{pt,mp} B_\mu(I_\beta(Have(\beta, screw))) \quad (\text{M.11})$$
$$\{(\text{M.7}),(\text{G.1})\} \vdash_{mp} B_\mu(Give(\alpha, Y, screw) \rightarrow Have(Y, screw)) \quad (\text{M.12})$$
$$\{(\text{M.11}),(\text{M.12}),(\text{G.2})\} \vdash_{pt,mp} B_\mu(I_\beta(Give(\alpha, \beta, screw))) \quad (\text{M.13})$$
$$\{(\text{M.13})\} \vdash_{R.1} Tell(\mu, \beta, I_\beta(Give(\alpha, \beta, screw))) \quad (\text{M.14})$$
$$\{(\text{M.14})\} \vdash_{R.3} I_\beta(Give(\alpha, \beta, screw)) \quad (\text{M.15})$$

The second argument is: $(I_\beta(Give(\mu, \beta, screwdriver)), P''_\beta)$, where P''_β is

$$\{(\text{M.2}),(\text{M.5}),(\text{G.2})\} \vdash_{pt,mp} B_\mu(I_\beta(Have(\beta, screwdriver))) \quad (\text{M.16})$$
$$\{(\text{M.1}),(\text{G.1})\} \vdash_{mp} B_\mu(Give(\mu, Y, screwdriver) \rightarrow Have(Y, screwdriver)) \quad (\text{M.17})$$
$$\{(\text{M.16}),(\text{M.17}),(\text{G.2})\} \vdash_{pt,mp} B_\mu(I_\beta(Give(\mu, \beta, screwdriver))) \quad (\text{M.18})$$
$$\{(\text{M.18})\} \vdash_{R.1} Tell(\mu, \beta, I_\beta(Give(\mu, \beta, screwdriver))) \quad (\text{M.19})$$
$$\{(\text{M.19})\} \vdash_{R.3} I_\beta(Give(\mu, \beta, screwdriver)) \quad (\text{M.20})$$

These two arguments represent advices to β on how it can achieve its goal (B.1) that was communicated to mediator μ as (M.5). Using bridge rule (R.4) β converts this into the following actions:

$$\{\text{M.15}\} \vdash_{R.4} Ask(\beta, \alpha, Give(\alpha, \beta, screw)).$$
$$\{\text{M.20}\} \vdash_{R.4} Ask(\beta, \mu, Give(\mu, \beta, screwdriver)).$$

When both α and μ receive this request, they convert this into *accept request* action using bridge rule (R.5). Mediator accepts this request due to the *generosity* theory (G.3), which defines that it is not an intention of mediator to own anything. Agent β cannot find a counter-argument that would reject this request (it does not need the nail) and accepts it. With the screw, the screwdriver, the mirror and knowledge on how to hang the mirror using these tools, β can fulfil its goal, and it no longer needs the nail. Therefore, the following argument that solves the goal of α is also accepted: $(I_\alpha(Give(\beta, \alpha, nail)), P_\alpha)$, where P_α is:

$$\{(\text{M.3}),(\text{M.4}),(\text{G.2})\} \vdash_{mp} B_\mu(I_\alpha(Have(\alpha, nail))) \quad (\text{M.21})$$
$$\{(\text{M.9}),(\text{G.1})\} \vdash_{mp} B_\mu(Give(\beta, Y, nail) \rightarrow Have(Y, nail)) \quad (\text{M.22})$$
$$\{(\text{M.21}),(\text{M.22}),(\text{G.2})\} \vdash_{pt,mp} B_\mu(I_\alpha(Give(\beta, \alpha, nail))) \quad (\text{M.23})$$
$$\{(\text{M.23})\} \vdash_{R.1} Tell(\mu, \alpha, B_\mu(I_\alpha(Give(\beta, \alpha, nail)))) \quad (\text{M.18})$$
$$\{(\text{M.18})\} \vdash_{R.3} I_\alpha(Give(\beta, \alpha, nail)) \quad (\text{M.19})$$

we convert this into action, using the bridge rule $R.1$ into:

$$\{\text{M.19}\} \vdash_{R.1} Ask(\alpha, \beta, Give(\beta, \alpha, nail)).$$

When agent β receives this request, β can accept it by the bridge rule (R.5). This is only possible because of the previous two arguments, when an alternative plan to hang the mirror was presented to β, otherwise β would not be willing to give up the nail needed for his plan. Agent β can now decide between two plans using (G.7); therefore it decides to *give* α the nail and both agents were able to fulfil their goals (we assume that β does not want to sabotage the mediation).

[2] *mp* stands for *modus ponens* and *pt* stands for *particularization*

6 Conclusion

Mediation brings more information and knowledge to the negotiation table, hence, an automated mediator would need the machinery that could do that. Addressing this issue in an automated setting, we have presented an ABN approach that extends the logic-based approach to ABN involving BDI agents, presented in [1]. We have introduced a mediator in the multiagent architecture, which has extensive knowledge concerning mediation cases and access to resources. Using both, knowledge and resources, the mediator proposes solutions that become the subject of further negotiation when the agents in conflict cannot solve the dispute by themselves. We have described our mediation algorithm and illustrated it with the same case study introduced in [1]. The presence of a mediator in ABN allows to deal with realistic situations when negotiation is stalled. In this work we assumed that the agents and the mediator operate within the same ontology, describing the negotiation domain. In real settings, the negotiators may interpret the same term differently. In order to avoid this, mediation will require the initial alignment of the ontologies with which all parties operate.

References

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

[2] Kolodner, J.L., Simpson, R.L.: The mediator: Analysis of an early case-based problem solver. Cognitive Science **13**(4) (1989) 507–549

[3] Sycara, K.P.: Problem restructuring in negotiation. Management Science **37**(10) (1991) 1248–1268

[4] Wilkenfeld, J., Kraus, S., Santmire, T.E., Frain, C.K.: The role of mediation in conflict management: Conditions for successful resolution. In: Multiple Paths to Knowledge in International Relations. Lexington Books (2004)

[5] Rahwan, I., Pasquier, P., Sonenberg, L., Dignum, F.: A formal analysis of interest-based negotiation. Annals of Mathematics and Artificial Intelligence **55**(3-4) (2009) 253–276

[6] Chalamish, M., Kraus, S.: AutoMed: an automated mediator for multi-issue bilateral negotiations. Autonomous Agents and Multi-Agent Systems **24** (2012) 536–564

[7] Lin, R., Gev, Y., Kraus, S.: Bridging the gap: Face-to-face negotiations with automated mediator. IEEE Intelligent Systems **26**(6) (2011) 40–47

[8] Simoff, S.J., Debenham, J.: Curious negotiator. In Klusch, M., Ossowski, S., Shehory, O., eds.: Proceedings of the Int. Conference on Cooperative Information Agents, CIA-2002, Springer, Heidelberg (2002)

[9] Bellucci, E., Zeleznikow, J.: Developing negotiation decision support systems that support mediators: case study of the Family_Winner system. Artificial Intelligence and Law **13**(2) (2005) 233–271

[10] Abrahams, B., Bellucci, E., Zeleznikow, J.: Incorporating fairness into development of an integrated multi-agent online dispute resolution environment. Group Decision and Negotiation **21** (2012) 3–28

[11] Visser, W., Hindriks, K.V., Jonker, C.M.: Interest-based preference reasoning. In: Proceedings of International Conference on Agents and Artificial Intelligence ICAART2011. (2011) 79–88

[12] Schei, V., Rognes, J.K.: Knowing me, knowing you: Own orientation and information about the opponent's orientation in negotiation. International Journal of Conflict Management **14**(1) (2003) 43–60

[13] Debenham, J.: Bargaining with information. In Jennings, N.R., Sierra, C., Sonenberg, L., Tambe, M., eds.: Proceedings Third International Conference on Autonomous Agents and Multi Agent Systems AAMAS-2004, ACM Press, New York (2004) 664–671

[14] Debenham, J.K., Simoff, S.: Negotiating intelligently. In Bramer, M., Coenen, F., Tuson, A., eds.: Proceedings 26th International Conference on Innovative Techniques and Applications of Artificial Intelligence, Cambridge, UK (2006) 159–172

[15] Sierra, C., Debenham, J.: Information-based agency. In: Proceedings of Twentieth International Joint Conference on Artificial Intelligence IJCAI-07, Hyderabad, India (2007) 1513—1518

[16] Bench-Capon, T.J.M.: Persuasion in practical argument using value-based argumentation frameworks. Journal of Logic and Computation **13**(3) (2003) 429–448

[17] Sycara, K.P.: Persuasive argumentation in negotiation. Theory and decision **28**(3) (1990) 203–242

[18] Kakas, A., Moraitis, P.: Adaptive agent negotiation via argumentation. In: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, ACM (2006) 384–391

[19] Rahwan, I., Ramchurn, S.D., Jennings, N.R., Mcburney, P., Parsons, S., Sonenberg, L.: Argumentation-based negotiation. The Knowledge Engineering Review **18**(04) (2003) 343–375

[20] Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence **77** (1995) 321–358

[21] Hadidi, N., Dimopoulos, Y., Moraitis, P.: Argumentative alternating offers. In: Argumentation in Multi-Agent Systems. Springer (2011) 105–122

[22] Phan Minh Dung. Phan Minh Thang, F.T.: Towards argumentation-based contract negotiation. Computational Models of Argument: Proceedings of Comma 2008 **172** (2008) 134

[23] Dijkstra, P., Prakken, H., de Vey Mestdagh, K.: An implementation of norm-based agent negotiation. In: Proceedings of the 11th international conference on Artificial intelligence and law, ACM (2007) 167–175

[24] Krause, P., Ambler, S., Elvang-Goransson, M., Fox, J.: A logic of argumentation for reasoning under uncertainty. Computational Intelligence **11**(1) (1995) 113–131

[25] Prakken, H., Sartor, G.: Argument-based extended logic programming with defeasible priorities. Journal of applied non-classical logics **7**(1-2) (1997) 25–75

[26] Dung, P.M., Kowalski, R.A., Toni, F.: Dialectic proof procedures for assumption-based, admissible argumentation. Artificial Intelligence **170**(2) (2006) 114–159

[27] Oren, N., Norman, T.J., Preece, A.: Subjective logic and arguing with evidence. Artificial Intelligence **171**(10) (2007) 838–854

[28] Amgoud, L., Dimopoulos, Y., Moraitis, P.: A unified and general framework for argumentation-based negotiation. In: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems, ACM (2007) 963–970

[29] Rahwan, I., Simari, G., eds.: Argumentation in Artificial Intelligence. Springer-Verlag (2009)

[30] Rao, A.S., Georgeff, M.P.: Formal models and decision procedures for multi-agent systems. Technical Report Technical Note 61 (1995)