Multiparty Argumentation Game for Consensual Expansion

Stefano Bromuri[1] and Maxime Morge[2]

[1]*Institute of Business Information Systems, University of Applied Sciences Western Switzerland, Sierre, Switzerland*

[2]*Laboratoire d'Informatique Fondamentale de Lille, Université Lille 1, Lille, France*

Keywords: Multi-Agent Systems, Argumentation, Multiparty Communication.

Abstract: We consider here a set of agents, each of them having her own argumentation. Arguments and conflicts between them are subjective. The aim of each agent is to enrich her argumentation by taking into account the arguments and conflicts of the other agents. We adopt here an individual-based approach where the cross-fertilization of argumentations emerge from the interactions between the agents. For this purpose, we formalize a multi-party argumentation game using Event Calculus. At the end of the game, each agent extends its argumentation by using the arguments exchanged and the conflicts shared. As we show formally, such an expansion is consensual. By adopting an individual-based approach, our model is explanatory since it highlights the conflicts between the agents.

1 INTRODUCTION

Multi-Agent Systems (MAS) is a first-class paradigm for analysing, designing and implementing systems composed of autonomous interacting entities. Oppositions characterize these systems. Conflicts may appear between agents since each of them has a partial perception of the environment or they have their own objectives. Well-known mechanisms such as voting systems or negotiation resolve these conflicts by aggregating the viewpoints to reach an agreement. Unfortunately, these models are not explanatory. In the argumentation approach, the positions (arguments) and the oppositions (attacks) are first-class citizens. Argumentation is a computational calculus of oppositions (Dung, 1995). In this paper, we adopt a dialectical approach of argumentation where the argumentation is the outcome of a dispute process (Prakken, 2006).

We consider here a set of agents, each of them having her own argumentation. Arguments and conflicts between arguments are subjective. They may be different. For example, an agent can consider that an argument attacks a second one while another agent may have the same arguments without considering them conflicting. Similarly, a third agent may not know one of these arguments and therefore she ignores this conflict.The aim of each agent is to enrich her argumentation by taking into account the arguments and conflicts of the other agents.

We adopt an individual-based approach where the cross-fertilization of argumentations emerge from the interactions between the agents. For this purpose, we formalize a multiparty argumentation game using Event Calculus (EC) (Kowalski and Sergot, 1986). At the end of the game, each agent extends its argumentation by using the arguments exchanged and the conflicts shared. As we show formally, such an expansion is consensual. By adopting an individual-based approach, our model is explanatory since it highlights the conflicts between the agents.

We start this paper by introducing the background formalisms on which we built our argumentation game (cf Section 2). Then, we define in Section 3, the consensual expansion which is the outcome of our game (cf Section 4). We discuss the related works (cf Section 5). Section 6 concludes by summarising our proposal and discussing our future plans.

2 BACKGROUND

In this section we discuss about the background formalisms that are necessary to understand our argumentation game.

2.1 Argumentation

In this paper we adopt the abstract approach to argumentation proposed in (Dung, 1995). Arguments are

Bromuri S. and Morge M..
Multiparty Argumentation Game for Consensual Expansion.
DOI: 10.5220/0004188301600165
In *Proceedings of the 5th International Conference on Agents and Artificial Intelligence* (ICAART-2013), pages 160-165
ISBN: 978-989-8565-38-9

viewed as abstract entities supporting claims. The fact that an argument is challenged by another captures the notion of conflict.

Formally, an *Argumentation Framework* (AF) is defined as follows.

Definition 1 (AF)**.** *An Argumentation Framework is a couple $AF = \langle A, R \rangle$ where A is a finite set of arguments, $R \subseteq A \times A$ is a binary relation called* attack relation.

If $(a,b) \in R$, then we say that a attacks b (we write this as aRb). We call *non-attack relation* $N = (A \times A) \setminus R$. In this way, we highlight that the lack of attack between arguments is interpreted as a non-attack relation.

An argumentation framework does not allow to model missing information. For this purpose, a *Partial Argumentation Framework* (PAF) is an extension of the AF model, proposed in (Coste-Marquis et al., 2007), where the fact that an argument attacks (or not) a second argument can be ignored. This missing information is captured by a binary relation, called ignorance relation.

Definition 2 (PAF)**.** *A partial argumentation framework is a triplet $PAF = \langle A, R, I \rangle$ where:*

- *$\langle A, R \rangle$ is the underlying AF as defined in Def. 1;*
- *$I \subseteq A \times A$ is the* ignorance relation *which verifies that $R \cap I = \emptyset$;*

We call *non-attack relation* $N = (A \times A) \setminus (R \cup I)$.

2.2 Event Calculus

The Event Calculus (EC), introduced in (Kowalski and Sergot, 1986), is a formalism for representing actions and their effects and so it is very suitable for formalising an interaction protocol for agents. In this section we briefly describe the dialect of the EC that we employ (Artikis et al., 2010).

EC is based on a many-sorted first-order predicate calculus represented as normal logic programs that are executable in Prolog. The underlying time model is linear. The EC manipulates fluents. A fluent represents a property which can have different values over time. The term $F = V$ (which denotes that fluent F has value V) has been initiated by an action at some earlier time-point and not terminated by another action in the meantime. Tab. 1 summarizes the main EC predicates we use in this paper. Predicates, function symbols and constants start with a lower-case letter while variables (starting with an upper-case letter) are universally quantified.

The domain independent axioms of the EC are represented in Fig. 1. Clause EC1 states that a property holds at a time `T` if it has been initiated at time `Ts` and the holding of that property has not been broken from the starting time `Ts` and the time of interest `T`. To decide when a property is broken, we use the clause EC2. This states that a property `P` is broken between time `Ts` and `T`, if it is terminated at a time `Ti` between `Ts` and `T`. The other clauses specify when a property is initiated EC3 or terminated EC4, in terms of the conditions holding in the current context, typically expressed in terms of the `holds_at/2` predicate, meaning that such clauses will change according to the particular domain being modelled with the EC.

Table 1: EC with multi-valued fluents: predicates.

Predicate	Meaning
initially(F=V)	The value of fluent F is V at time 0.
holds_at(F=V,T)	The value of fluent F is V at time T.
initiates_at(F=V, T)	At time T the fluent F is initiated to have value V.
terminates_at(F=V, T)	At time T the fluent F is terminated from having the value V.
update(E,T)	An event E takes place at time T updating the state of the fluents

```
(EC1) holds_at(F=V, T) ← initiates_at(F=V, Ts),
                         Ts < T,
                         not broken(F=V, Ts, T).
(EC2) broken(F=V1, Tmin, Tmax)← terminates_at(F=V2, Ti),
                         not V1 = V2,
                         Tmin < Ti, Ti < Tmax.
(EC3) initiates_at(F=V, T) ← happens_at(Ev,T),
                         Conditions[T].
(EC4) terminates_at(F=V, T)← happens_at(Ev,T),
                         Conditions[T].
```

Figure 1: Axioms of the EC with multi-valued fluents.

The EC allows us to construct executable specifications of interaction protocols for agents.

3 PROBLEM

We consider here a set of agents, each of them having its own arguments. The agents aim at expanding their arguments, in particular the conflicts between them, based on a consensus.

Formally, we consider here a profile of n argumentation frameworks $\vec{S} = \langle AF_1, \ldots, AF_n \rangle$. Our goal here is to expand this vector in a profile of partial argumentation frameworks $\vec{P} = \langle PAF_1, \ldots, PAF_n \rangle$ where each PAF_i expands the corresponding AF_i with $\vec{S}$ by taking account the arguments, the attacks and the non-attacks from the other AFs in $\vec{S}$.

For this purpose, we focus on the consensual expansion proposed by (Coste-Marquis et al., 2007). In order to expand an AF_i on a PAF_i, we consider all the arguments and only the consensual attacks.

Definition 3 (Consensual Expansion). *Let $\vec{S} = \langle AF_1, \ldots, AF_n \rangle$ be a profile of n argumentation frameworks $AF_i = \langle A_i, R_i \rangle$ (cf Def. 1) with $1 \leq i \leq n$. Let $conf(\vec{S}) = (\bigcup_i R_i) \cap (\bigcup_i N_i)$ be the set of non-consensual attacks. The consensual expansion of AF_i with $\vec{S}$ is a partial argumentation framework $PAF_i = \langle A', R', I' \rangle$ where:*

- $A'_i = \bigcup_i A_i$;
- $R'_i = R_i \cup ((\bigcup_{j \neq i} R_j) \setminus conf(\vec{S}))$;
- $I'_i = conf(\vec{S}) \setminus (A_i \times A_i)$.

Obviously, $N'_i = (A'_i \times A'_i) \setminus (R'_i \cup I'_i)$. The arguments in the consensual expansion (A'_i) are all the arguments from the initial profile. A new attack is added (R'_i) if it is consensual, i.e. if all agents which initially consider these arguments agree on this conflict. An attack is ignored (I'_i) if it is not consensual ($conf(\vec{S})$) and if it was not considered *a priori* ($A_i \times A_i$).

It is worth noticing that, in order to determine a consensual expansion, we assume to know the arguments and the attacks (and non-attacks) of all the agents. In the next section, we do not make such an assumption.

4 PROPOSAL

We adopt here an individual-oriented approach where the consensual expansion emerges from the interactions between the agents. Our proposal consists of a multiparty argumentation game, where more than two agents, each of them with a AF, play and observe moves on a gameboard. At the end of the game, each agent builds its PAF with the arguments and the conflicts recorded on the game-board.

Firstly, we begin to introduce the gameboard. Secondly, we formalize the argumentation game and its output. Then we prove the correctness of the game. Finally, we illustrate this game with an example.

4.1 Gameboard

The gameboard is a common environment perceived by agents in which they can act by adding arguments, attacks and non-attacks. These moves are evaluated via an artifact which records the dialogue history.

Definition 4 (gb). *At any moment, the gameboard is an objective representation of the game with a triplet $gb = \langle AM, RM, DM \rangle$ where:*

- *AM is the record list of argument moves;*
- *RM is the record list of attack moves;*
- *DM is the record list of denial moves.*

Each utterance of a move is interpreted by the artifact for updating the gameboard in order to build the common partial argumentation framework.

Definition 5 (AF_{gb}). *We call common partial argumentation framework $PAF_{gb} = \langle A_{gb}, R_{gb}, D_{gb} \rangle$, the argumentation framework defined such that:*

- *A_{gb} is the set of arguments in the argument moves of AM;*
- *R_{gb} is the set attacks in the attack moves of RM;*
- *D_{gb} is the set of ignorances in the denial moves of DM.*

We aim at defining the game such that the rational rules of utterances and the rules of the game leads to a common partial argumentation framework $PAF_{gb} = \langle A_{gb}, R_{gb}, D_{gb} \rangle$ which is:

1. **complete**, i.e. $A_{gb} = \bigcup_i A_i$;
2. a **coherent** partial argumentation framework, i.e. $R_{gb} \cap D_{gb} = \emptyset$;
3. **consensual**, i.e. R_{gb} (resp. D_{gb}) belongs all the attacks which are consensual (resp. non-consensual).

4.2 Argumentation Game

The game we propose is a n-players simultaneous game. We formalize here our game using the EC and the rational condition of utterance using abstract argumentation.

Our argumentation game is subdivided into two subgames. The first one aims at collecting all the arguments of the agents. It is called **argument game**. The second game aims at collecting all the consensual attacks and all the non-consensual ones. It is called **attack game**. The argument (resp. attack) argument ends when all the players withdraw, i.e. when they have no more arguments (resp. attacks) to push forward.

The two following clauses handle the creation of the game:

```
initiates_at(gb(Gb)=state(argGame),T):-
  happens_at(create_game(GID,PL),T),
  not holds_at(gb(Gb)=state(_),T).
initiates_at(gb(Gb)=players_list(PL),T):-
  happens_at(create_game(Gb,PL),T),
  not holds_at(gb(Gb)=state(_),T).
```

While the first clause handles the creation of the gameboard GID by setting the game state argGame, the second clause sets the list of players PL.

Argument Game. The moves update the gameboard only if they are legal moves. The update_game/3 predicate below, specifies that a Move is produced as an event that modifies the state of a game Gb only if such a move is legal at time T in such a game.

```
update_game(Gb,AID,Move):-
now(T), legal_at(Gb,AID,Move,T),
  update(Move,T).
```

The legal_at/4 predicate defines which moves can be performed at a certain time. For example, if the agent is trying to produce an assertion move, the artifact checks if the argument in the assertion is a new one. If this is the case it updates the common partial argumentation framework.

```
initiates_at(gb(Gb)=argument(AID,Arg),T):-
  happens_at(move(Gb,AID,all,assert,ArgID,Arg),T).
legal_at(Gb,AID,move(Gb,AID,all,assert,ArgID,Arg),T):-
  holds_at(move(Gb,AID,all,assert,ArgID,Arg),T),
  not (holds_at(gb(Gb)=argument(_,Arg),T)).
```

In the case that the argument is already present in the gameboard, the gameboard will not be updated, because the legal_at/4 predicate defined above will check if in the gameboard Gb it exists an argument Arg that is the same as the one being asserted. It is important to notice that the argument id (ArgID) is required as the gameboard can have multiple arguments simultaneously initiated. Furthermore, the legal_at/4 predicate can be used by the agents to check if the next move they want to perform is a legal move. In other words the legal_at/4 predicate can be used by the agents in combination with the holds_at/2 predicate of the EC to perceive the gameboard state.

The argument game ends when all the players withdraw:

```
terminates_at(gb(Gb)=state(argGame),T):-
  happens_at(move(Gb,AID,all,withdraw,no,no),T),
  not(
    holds_at(gb(Gb)=player(AID2),T),
    holds_at(player(AID2)=status(moreArg),T),
    not AID = AID2
  ).
initiates_at(gb(Gb)=state(attGame),T):-
  happens_at(move(Gb,AID,all,withdraw,no,no),T),
  not(
    holds_at(gb(Gb)=player(AID2),T),
    holds_at(player(AID2)=state(Gb,moreArg),T),
    not AID = AID2
  ).
```

In the rules defined above, the argument game ends when all the agents perform a withdraw move, and so the attack game is initiated. For the argument game, the withdraw move is always legal.

Attack Game. In this game, agents assert/deny attacks or withdraw. After an assertion (resp. deny), the artifact updates the common partial argumentation framework if it is the case:

```
initiates_at(gb(Gb)=attack(AttID),T):-
  happens_at(move(Gb,AID,all,assert,AttID,Att),T).
initiates_at(gb(Gb)=ignore(AttID),T):-
  happens_at(move(Gb,AID,AID2,deny,AttID,Att),T).
```

Similarly to the previous subgame, the moves update the gameboard only if they are legal moves.

```
legal_at(Gb,AID,move(Gb,AID,all,assert,AttID,Att),T):-
  not (
  holds_at(gb(Gb)=attack(_,Att),T),
  holds_at(gb(Gb)=ignore(_,Att),T)
  ).
legal_at(Gb,AID,move(Gb,AID,all,deny,AttID,Att),T),
  holds_at(gb(Gb)=attack(_,Att),T),
  not holds_at(gb(Gb)=ignore(_,Att),T).
```

The two legal_at/4 predicates specified above state that the assertion of an attack is legal if the attack is not already present in the game, even if the identifier is different. Similarly, the deny of an attack is legal if the attack already exists in the gameboard and the attack is not already ignored. As for the argument game, a withdraw move is always legal.

The attack game ends when all the players withdraws, which is also the termination condition for the whole game:

```
terminates_at(gb(Gb)=state(attGame),T):-
  happens_at(withdraw(Gb,AID),T),
  not(
    holds_at(gb(Gb)=player(AID2),T),
    holds_at(player(AID2)=status(more_attacks),T),
    not AID = AID2
  ).
initiates_at(gb(Gb)=state(final),T):-
  happens_at(withdraw(Gb,AID),T),
  not(
    holds_at(gb(Gb)=player(AID2),T),
    holds_at(player(AID2)=status(more_attacks),T),
    not AID = AID2
    ).
```

Rational Rules. Each player ag_i, which is associated with an argumentation framework $AF_i = \langle A_i, R_i \rangle$, can check the legality of moves using the EC predicate legal_at/4 and submit it if it is the case.

As previously discussed, during the argument game, agents assert new arguments or withdraw. At time t, an agent can submit a move based on the following rational rule:

$$ag_i \text{ utters } \begin{cases} m = \langle gid, ag_i, all, assert, a \rangle \\ \text{ if } \exists a \in A_i \wedge legal\_at(gid, ag_i, m, t) \\ m = \langle ag_i, all, withdraw, null \rangle \text{ else} \end{cases} \quad (1)$$

During the attack game, an agent can submit at time t a move based on the following rational rules:

$$ag_i \text{ utters } \begin{cases} m = \langle gid, ag_i, all, assert, (a,b)\rangle \text{ if} \\ \exists (a,b) \in R_i \wedge legal\_at(gid, ag_i, m, t) \\ m = \langle gid, ag_i, ag_j, deny, (a,b)\rangle \text{ if} \\ \exists (a,b) \in N_i \wedge legal\_at(gid, ag_i, m, t) \\ \langle gid, ag_i, all, withdraw, null\rangle \text{ else} \end{cases} \quad (2)$$

Here we assume that agents are honest.

4.3 Game Over

At the end of the game, some properties about the common partial argumentation framework hold. For brevity, we do not include here the proofs.

Property 1. *The following properties are verified:*

- *The argument game and the attack game end.*
- *At the end of the argument game,* $A_{gb} = \bigcup_i A_i$.
- *At each turn,* $R_{gb} \cap D_{gb} = \emptyset$.
- *At the end of the attack game,* $(a,b) \in R_{gb}$ *iff* $(\exists i \mid (a,b) \in R_i) \wedge (\nexists j \mid (a,b) \in N_j))$.
- *At the end of the attack game,* $(a,b) \in D_{gb}$ *iff* $\exists i, j \mid (a,b) \in R_i \wedge (a,b) \in N_j$.

4.4 Outcome

At the end of the game, each player expands her argumentation framework with the arguments, the attacks and the denials reported in the common partial argumentation framework:

Definition 6 (Game Expansion)**.** *Let* $\vec{S} = \langle AF_1, \ldots, AF_n\rangle$ *be a profile of* n *argumentation frameworks and* $AF_i = \langle A_i, R_i\rangle$ *be one of them. The expansion of* AF_i *with* $PAF_{gb} = \langle A_{gb}, R_{gb}, D_{gb}\rangle$ *is the partial argumentation framework* $PAF_i = \langle A_i'', R_i'', I_i''\rangle$ *defined such that:*

1. $A_i'' = A_{gb}$;
2. $R_i'' = R_i \cup R_{gb}$
3. $I_i'' = D_{gb} \setminus (A_i \times A_i)$

Obviously, $N_i'' = (A_i'' \times A_i'') \setminus (R_i'' \cup I_i'')$. It is worth noticing that this expansion is built upon the initial argumentation framework of the agent and the common partial argumentation framework. Contrary to the consensual expansion (cf Def. 3), we do not need to know the arguments and attacks of all the other agents but only the outcome of the game.

The common partial argumentation framework allows to expand the individual argumentation frameworks in a consensual manner.

Property 2 (Consensual Game Expansion)**.** *Let* $\vec{S} = \langle AF_1, \ldots, AF_n\rangle$ *a profile of* n *argumentation framework and let* $AF_i = \langle A_i, R_i\rangle$ *be one of them. The expansion of* AF_i *with the common partial argumentation framework* $PAF_{gb} = \langle A_{gb}, R_{gb}, D_{gb}\rangle$ *is consensual (cf Def. 3).*

4.5 Example

We present here an illustrative example for our game.

We consider four agents associated to the profile of argumentation frameworks depicts at top of Fig. 2. At the end of the argument game, the common argumentation framework contains all the existing arguments in the system, $A_{gb} = \{a,b,c,d\}$. At the end of the attack game, the common argumentation framework (represented in the middle of Fig. 2) contains all the consensual attacks ($R_{gb} = \{(b,c),(c,d)\}$) represented by plain arrows and all the non-consensual attacks represented by dotted arrows ($D_{gb} = \{(b,d),(b,a),(a,b),(a,d)\}$). At the end of the game, each agent build her own PAF (shown at the bottom of Fig. 2).

Finally, the common argumentation framework contains all the conflicts exhibited through the dialogue in D_{gb}. Moreover, the gameboard contains in *DM* all the conflicts which arise between the agents:

- the agents ag_3 and ag_4 disagree on (a,b), $m_6 = \langle ag_4, ag_3, deny, (a,b)\rangle$;
- the agents ag_2 et ag_4 disagree on (b,d), $m_9 = \langle ag_2, ag_4, deny, (b,d)\rangle$;
- the agents ag_1 et ag_3 disagree on (b,a), $m_{10} = \langle ag_3, ag_1, deny, (b,a)\rangle$;
- the agents ag_3 and ag_4 disagree on (a,d), $m_{12} = \langle ag_4, ag_3, deny, (a,d)\rangle$;

It is worth noticing that this list of disagreements is not exhaustive. For instance, this execution does not underline neither the disagreement between the agents ag_1 and ag_4 over the attacks (b,a), nor the disagreement between the agents ag_1 and ag_4 over the attacks (a,b).

5 RELATED WORKS

There is few research on multiparty argumentation games. Recently, (Bonzon and Maudet, 2011) have proposed a multiparty persuasion protocol. This work differs from our proposal. On one hand, (Bonzon and Maudet, 2011) assume that all agents share the same arguments and only the attacks are subjective. We consider here that the arguments are distributed in

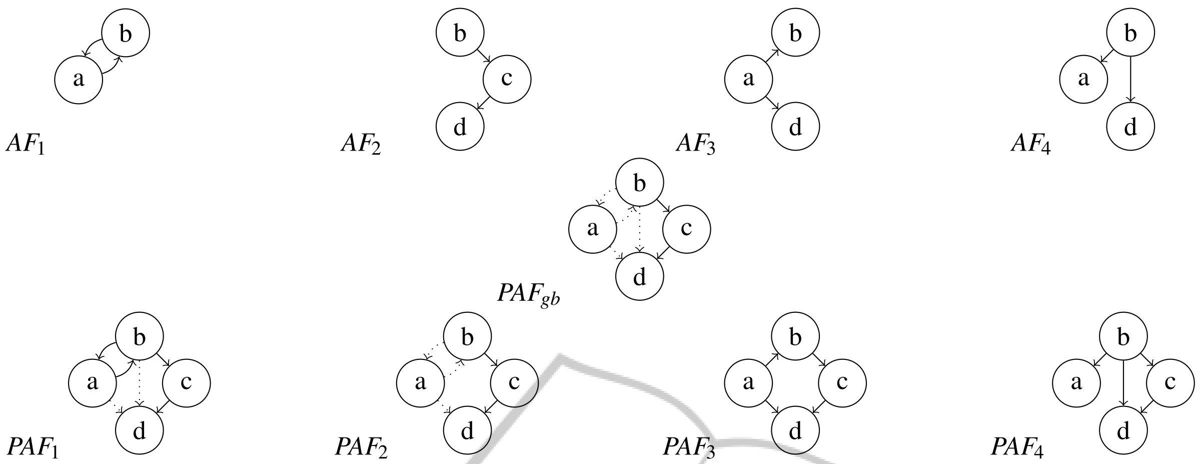

Figure 2: The profile of argumentation frameworks (at top), the common argumentation framework (at middle) and the profile of partial argumentation frameworks (at bottom).

the multi-agent system. On the other hand, the game proposed by (Bonzon and Maudet, 2011) arbitrates among conflicting viewpoints. The outcome of such an arbitration is then compared with the outcome of the fusion of these viewpoints. In our approach, the objective of the game is not a collective decision but a cross-fertilization of views and the detection of the conflicts between the agents.

(Bodenstaff et al., 2006) study the suitability of the Event Calculus for formalising argumentation game and its implementation with a declarative programming language by carrying out two case studies. With respect to this work, we differ as we consider a multiparty game and our agents use the declarative formalisation of the game to reason about the moves which are allowed and the agents choose one of them.

6 CONCLUSIONS

In this paper, we have considered a set of agents, each of them is equipped with her own argumentation framework. In this way, the arguments are distributed in the multi-agent system and the conflicts between theses arguments are subjective. We have formalized a multiparty argumentation game, where more than two agents play and observe moves on a gameboard. In this dialogue, each agent aims at expanding her argumentation by taking into account all the available arguments and the consensual conflicts between these arguments. In our individual-oriented approach, the building of the consensual expansions emerges from the interactions between the agents. We demonstrated the termination and the correctness of our game. The expansions are consensual if the agents are honest. Moreover, our model is explanatory since it renders intelligible the conflicts between the agents which appear during the process.

REFERENCES

Artikis, A., Sergot, M., and Paliouras, G. (2010). A logic programming approach to activity recognition. In *ACM International Workshop on Events in Multimedia*, pages 3–8.

Bodenstaff, L., Prakken, H., and Vreeswijk, G. (2006). On formalising dialogue systems for argumentation in the event calculus. In *Proceedings of the Eleventh International Workshop on Nonmonotonic Reasoning*, pages 374–382.

Bonzon, E. and Maudet, N. (2011). On the outcomes of multiparty persuasion. In *Proc. of the 10th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pages 47–54.

Coste-Marquis, S., Devred, S., Konieczny, S., Lagasquie-Schiex, M.-C., and Marquis, P. (2007). On the merging of Dung's argumentation systems. *Artif. Intell.*, 171:730–753.

Dung, P. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77:321–357.

Kowalski, R. and Sergot, M. (1986). A logic-based calculus of events. *New Gen. Comput.*, 4(1):67–95.

Prakken, H. (2006). Formal systems for persuasion dialogue. *The Knowledge Engineering Review*, 21:163–188.