

Argumentation and Artifact for Dialogue Support

Enrico Oliva¹

Mirko Viroli Andrea Omicini Peter McBurney

¹ALMA MATER STUDIORUM—Università di Bologna, Cesena, Italy

Argumentation in Multi-Agent Systems (ArgMAS)
12th, May 2008



Outline

1 Motivation/Background

- Motivation
- Background
- Mediator Functionalities

2 Solution/What we have done

- Overview
- Argumentation and Dialog Systems
- Dialog Artifact
- Implementation of Dialogue Artifacts
- Case Study

3 Conclusions

- Future Work



Argumentation Theory

Actually Argumentation is

- a formal discipline within Artificial Intelligence where the aim is to make a computer assist in or perform the act of argumentation

Argumentation is useful

- where formal logic and classical decision theory are unable to capture the richness of reasoning
 - in complex software systems to represent conflicting knowledge
 - in the construction of systems for legal reasoning
 - in Multi-agent Systems to model the **communication** between agents



...**good communication** can help to overcome and resolve almost any problems!



Argumentation and Dialogue

Dialogue is a reciprocal conversation between two or more people/agents

- could be modelled essentially as a dialectical exchange of arguments

Argumentation supports dialogue in MAS in order to

- exchange information, resolve disputes and persuade each other

[Walton and Krabbe, 1996] defines six types of dialogues among two entities: persuasion, inquiry, negotiation, information seeking, deliberation and eristic



Mediation and Dialogue

The role of mediator in a human interaction

- may act to ensure fairness and equality of access by all participants
- may assist participants to clarify their positions and to argue more effectively
- may even seek to reconcile opposing views

Just as in human society the mediators could be useful in MAS when software agents engage argumentative dialogues

- to avoid articulated process of rational deliberation



Objectives

- to provide an intelligent mediator service for dialogue support
- to extend the dialogue concept from two agents to N agents providing an infrastructure and theoretical support

To this end we extend the conceptual framework for a central co-ordinating entity, called *Co-Argumentation Artifact (CAA)* introduced by [Oliva et al, ArgMAS 2007]

- to enrich the dialogue protocol with argumentative reasoning operation
- to define a unified approach for communication and reasoning with argument

To this end we define and implement a dialogue system connected with argumentative reasoning operation

Multi-agent Systems

MAS is composed by autonomously interacting and heterogeneous entities

- A&A meta-model for MAS proposes two fundamental abstractions: agents and artifacts [*Omicini et al, J.AAMAS 2008*]

Agent is an autonomous and proactive computational entity situated in an environment

Artifact is a reactive non-autonomous computational entity that is used by agents, and it provides a fully automatized service.

Our model of a Multi-agent Argumentation System follows the A&A meta-model



Dialogue Artifact Functionalities

Artifacts for argumentation could be provide the support of basic and advanced functionality for ***automatic mediation services***

Some basic functionality to support the exchange of arguments in a dialogue between the participants includes:

- ➊ Storage of the dialogue protocol (e.g. in a library of such protocols)
- ➋ Storage of the specifications of the dialogue protocol
- ➌ Storage of the complete history of a dialogue as it proceeds
- ➍ The ability to refuse to allow agent utterances which do conform to the current protocol in use
- ➎ The ability to suggest next moves which are legal according to the current protocol in use in a dialogue
- ➏ The ability to receive and store confidential information from the participating agents, such as their preferences in a negotiation
- ➐ Supporting multiple simultaneous bilateral interactions

Overview

Map of our work

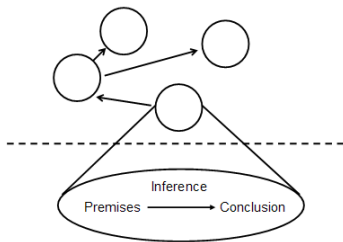
Theory	Argumentation System (AS)	Dialogue System (DS)
Model	Co-Argumentation Artifcat (CAA)	Dialogue Artifact (DA)
Code	TuCSon	TuCSon
Case Study	Argument Acceptance	Information Seeking



Our Argumentation Approach

The idea

Arguments are built using a monotonic logic (with first order language) and non-monotonicity is expressed at the meta level in terms of interaction between conflicting arguments



- Abstract Argumentation framework [Dung et al, 1995] - meta level
- Internal structure of argument express with first order language with tuple notation



Argument Representation

Argument is a sequence of inferences that leads to a conclusion. It is composed of

- **beliefs** are facts and rules that represent premises
- **inference rules** are labels that represent inference processes such as deduction or induction
- **conclusions** are facts that represent results of the inference process applied to the beliefs

Socrates Argument example

all men are mortal, Socrates is a man $\vdash_{MP}$ *Socrates is mortal*
`arg(name, beliefs([human(Socrates)], [clause(mortal(X)
, [human(X)])]), inf(MP), conclusion([mortal(Socrates)])) .`



Argumentation System

Argument is a minimal set of facts that leads to a conclusion through a sequence of inferences

- premises $\vdash$ conclusion

Attacks (or defeat) among arguments are

- **Rebuttal** ($\leftrightarrow$)
 - premise1 $\vdash$ conclusion1
 - premise2 $\vdash$ (not) conclusion1
- **Undercut** ($\rightarrow$)
 - premise2 $\vdash$ (not) premise1

Acceptability of an argument follows the notion of abstract Dung framework



Argumentation-based Dialogue System

Argumentation-based dialogue system is composed of

- a communication language
- a dialogue protocol
- (a protocol semantics)

Argumentation is used to model the content of communication and drive the dialog among agents in a MAS



Communication Language

Our *communication language* is a set of locutions L_c .

Definition (Locution)

A locution $l \in L_c$ is an expression of the form $perf_{name}(Arg_1, \dots, Arg_n)$ where $perf_{name}$ is an element of the set P of performatives and Arg_x is either a fact or an argument.

An agent can perform a locution composed of

- *facts* with syntax `fact (Terms)`
- *arguments* with syntax `argument (B, I, C)`

Information seeking dialog

Set of moves, identified as “locutions”

`OpenDialog, Ask, Tell, DontTell, Provide, Argue`



Dialogue Protocol

In our framework the dialogue protocol is

- a complete description of all dialogue paths
- a step by step description of the mediator behaviour

Technically our model is formalized using a *process algebra*

Definition (action)

An *action* A has the syntax $A ::= s : L_c | s[t_1, \dots, t_n] : L_c$ where s indicates the source, and $[t_1, \dots, t_n]$ indicates the (optional) targets

Definition (Term Action)

A *term action* K has the syntax $K ::= in(C, X) | out(C, X) | rd(C, X)$, where C represents commitment store identifier, and X represents the commitment

Definition (Protocol)

A *protocol* P is a composition of action from sets A and K , defined by syntax $P ::= 0 | A.P | K.P | P + P | (P \parallel P) | !P$



Example of dialogue protocol

- Set of moves, identified as “locutions”
 - e.g. *s:assert*, *s:why*, *s:accept*, *s:reject*, *s:retract*, *s:argue*
- Set of commands to manipulate commitment store
 - *in(C,t)*, *rd(C,t)*, *out(C,t)*

Specification of Information Seeking Dialogue

```

c:Opendialogue.
s:Opendialogue.
c:Ask(f). (
    rd(permission(c,f)).
    s:Tell(f).
    s:Provide(f).
    s:Argue(permission(c,f), YES, A).
    s:Enddialog.
    c:Enddialog
+
    rd(not(permission(c,f))).

```



Protocol Semantics

Protocol semantic is expressed with an operational semantic considering Charles Hamblin's notion of commitment stores

Definition (operational semantics)

The *operational semantics* is described by a **Labelled Transition System** $\langle S, \rightarrow, I \rangle$, where

- $S ::= (C)P$ represents the state of dialogue system (protocol P running with commitment store C)
- I is the set of interactions (labels) composed of $i ::= \tau | a$
- $\rightarrow$ is a transition relation of the kind $\rightarrow \subseteq S \times I \times S$

$s \xrightarrow{i} s'$ in place of $\langle s, i, s' \rangle \in \rightarrow$ means the dialogue system moves from state s to s' due to

- an action a
- an internal step τ (operation over commitment store)



Operational Semantic

Operational rules that describe the behaviour of the dialogue system:

$$\begin{array}{llll}
 (C)out(x).P & \xrightarrow{\tau} & (C|x)P & (K - OUT) \\
 (C|x)rd(y).P & \xrightarrow{\tau} & (C|x)P\{x/y\} & (K - RD) \\
 (C|x)in(y).P & \xrightarrow{\tau} & (C)P\{x/y\} & (K - IN) \\
 (C)(P + Q) & \xrightarrow{i} & (C')P' & \text{if } (C)P \xrightarrow{i} (C')P' \quad (OP - SUM) \\
 (C)(P|Q) & \xrightarrow{i} & (C')(P'|Q) & \text{if } (C)P \xrightarrow{i} (C')P' \quad (OP - PAR) \\
 (C)a.P & \xrightarrow{a'} & (C)P\{a'/a\} & (ACT)
 \end{array}$$



Dialog Artifact I

Dialog Artifact (DA) as a mediator of communication between agents that engaged a dialog under the argumentation dialog system.

Definition (Dialogue Artifact)

A *Dialogue Artifact* is a triple $DA = \langle DP, CS, IC \rangle$, where

- DP is a collection of specifications of dialogue protocols
- CS is a collection of commitment stores
- IC is a collection of specifications of interaction control (IC)



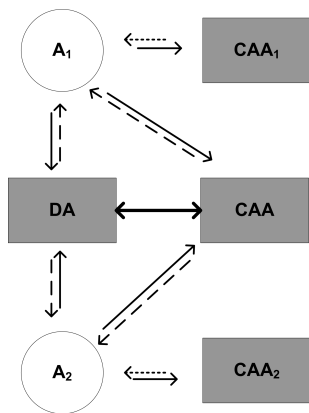
Dialog Artifact II

DA is a triple **DP,CS,IC**, where:

- DP is a collection of specifications of dialogue protocols
 - agents make utterances according to the permitted sequences defined by the protocol specification
- CS are a collection of commitments stores
 - they could be private and public of each participant, together with a central for the dialogue as a whole
- IC is a collection of specifications of interaction control
 - it identifies which constraints on the future course of dialogues are created by the existing commitments
 - it realizes the operator $next'(s) = \left\{ i : s \xrightarrow{i} s' \right\}$ in order to identify which utterances are legal



Architecture



We propose a scalable architecture for the agent society composed of

- local CAA private for an agent to coordinate its mental state
- global DA and CAA common for agent society
 - to provide services for coordination and communication based on arguments



Implementation

Technological support to realize the DA is provided by TuCSoN a coordination infrastructure for MAS

TuCSoN provides *programmable tuple spaces* where the agents can read/write and consume logic tuples

Argumentation process is composed of

- knowledge representation
- computation over argument sets

TuCSoN infrastructure supports

- knowledge declaratively represented in term of logic-tuple arguments
- computation over argument set in term of ReSpecT specification tuples

DA are built as a programmed tuple centres



DA Implementation I

DA is build as TuCSoN tuple centre where ReSpecT reactions implements the control of dialogue interaction

- DP is composed in terms of tuples `participant`, `dialogue`, `dialoguestate`, `currentpar`, `session`
- CS is realized by several CAA
- IC (the engine of process algebra management) is implemented exploiting a transition system defined by the predicates `transition(Currentstate, Action, Newstate)`.

```
transition([Act|S], A, S) :- transition(Act, A, zero).
transition(S1+S2, A, R1) :- transition(S1, A, R1).
transition(S1+S2, A, R2) :- transition(S2, A, R2).
reaction(out(move(Dialog, Id, Act)), ( %Start reaction
in_r(dialogstate(Dialog, S)),
out_r(transition(S, act(Id, Act), C, Dialog)) ) ).
...
```



DA Implementation II

Implementation of *next*^I operator in ReSpecT

```
%reacts from agent next moves request
reaction(rd(nextmoves(Dialogue,S)), (
    rd_r(dialoguestate(Dialogue,S)),
    out_r(findall(S,Dialogue))
)).
reaction(out_r(findall(S,Dialogue)), (
    in_r(findall(S,Dialogue)),
    findall(A,transition(S,A,Q),L),%collect all next legal moves
    out_r(nextmoves(Dialogue,L))
)).
```



A possible Case Study ADR/ODR

Alternative Dispute Resolution (ADR) is an alternative to litigation (negotiation, mediation, collaborative law, and arbitration)

- Arguments have a central role in the process of formal legal systems and also in the trial
- Disputants use arguments in order to persuade the other parts: such as decision makers - juries, judges, clients and attorneys
- **Online Dispute Resolution** is an online instance of ADR system moving it to virtual environment and providing computation and communication support



A possible Case Study ADR/ODR

The idea

We aim to exploit our framework **CAA and DA** such as computation and communication support for conflict resolution in an agent-based society

- In the DA we store the arbitration, mediation or negotiation protocol. The parties exploit the DA to take part in the discussion, which drives the dialogue grounded on the commitments
 - The advantages are: the management of dialogue between multiple entities and the automatic interaction with commitment/argument store
- The CAA provides the right abstraction to make a commitment/argument store where it is possible to evaluate automatically the argument validity respect with normative context



Conclusions I

At practical level

All six basic functionalities of the central Mediator can be performed by a Dialogue Artifact

- The collection *DP* provides the functionalities of
 - the storage of protocols (items 1)
 - the storage of protocols formal specifications (item2)
- The collection *CS* provides storage
 - for the history of a dialogue (item 3)
 - for confidential information (item 6)
- The rules in *IC* allow the DA
 - to suggest possible legal next moves (item 5);
 - to identify utterances which do not conform to the protocol (item 4)



Conclusions II

At the model level

- We proposed a model for coordination and reasoning in an agent society that merges concepts from argumentation and artifact theories
- DA and CAA are a framework to make argumentation and dialectical agent interaction to an operative level



Future Work

Inspired from the new perspective introduced by [*Rahwan et al, 2008*] that lays the basis for World Wide Argument Web WWAW

- We aim to generalize our argument representation with the AIF ontology and with argument based on presumptive inference
- We aim to obtain a tool for a World Wide Argumentation Web based on artifacts where agents and humans could indistinctly interact

Following the case study of ADR / ODR

- We aim to complete the framework with a better formalization of the relation between DA and CAA
- We aim to improve the protocol operations in order to facilitate the dialogue specification



Thank you!

Questions?

