International Journal of Artificial Intelligence,
Spring 2010, Volume 4, Number S10
ISSN 0974-0635;

Argumentation and Artifacts for Negotiation Support

Enrico Oliva[1], **Peter McBurney**[2], **Andrea Omicini**[1] **and Mirko Viroli**[1]

[1]Alma Mater Studiorum–Università di Bologna
via Venezia 52, 47023 Cesena, Italy
enrico.oliva, andrea.omicini, mirko.viroli @unibo.it

[2] Department of Computer Science, University of Liverpool
Ashton Building, Liverpool L69 3BX UK
mcburney@liverpool.ac.uk

ABSTRACT

*Negotiation is a central process in an agent society where autonomous agents have to cooperate in order to resolve conflicting interests and yet compete to divide limited resources. A direct dialogical exchange of information between agents usually leads to competitive forms of negotiation where the most powerful agents win. Alternatively, an intelligent mediated interaction may better achieve the goal of reaching a common agreement and supporting cooperative negotiation. In both cases argumentation is the reference framework to rationally manage conflicting knowledge or objectives, a framework which provides the fundamental abstraction "argument" to exchange pieces of information. In this paper we present a novel conceptual framework for negotiation dialogues using argumentation between autonomous software agents which enables their dialogues to be automated. The framework, called **SANA** (Supporting Artifacts for Negotiation with Argumentation), incorporates intelligent components able to assist the agent participants to reach agreement by inferring mutually-acceptable proposals. The framework also permits agents to engage in negotiation dialogues with each other, generating and exchanging proposed deals and arguments for and against these proposals. Acceptability of proposals is then assessed in terms of an agreed argumentation framework semantics. We present the architecture of our framework, along with the syntax, and outline denotational semantics of an associated agent interaction protocol, called **SANAP**.*

Keywords: Agents, Argumentation, Artifacts, Dialogues, Multiagent systems, Negotiation, Logic Programming.

2000 Mathematics Subject Classification: 68T35, 68T37, 68T42.

1 Introduction

A society mainly evolves through interaction and communication among participating entities. Within a society, people argue and negotiate in order to solve problems, to resolve or reduce conflicts, to exchange information, and to inform each other of pertinent facts. In particular, argumentation is a useful feature of human intelligence that enables us to deal with incomplete

and inconsistent information. People usually have only partial knowledge about the world (they are not omniscient) and often they have to manage conflicting information.

In the same way, the entities that compose an artificial society should be able to deal with partial and conflicting knowledge. Correspondingly, an agent-based model for an artificial society should provide adequate support for argumentation with the purpose of providing a realistic reflection of a society, providing means to share information in order to successfully deal with partial knowledge.

This work presents a conceptual framework for negotiation dialogues based on argumentation called **SANA** (Supporting Artifacts for Negotiation with Argumentation). The main form of communication to resolve conflict in human and artificial society is negotiation. Concretely, negotiation is an argumentative process where the participants compete for limited resources or collaborate to find common agreement over their division or allocation.[1] In the context of multi-agent systems there exist several approaches to realise automated forms of negotiation, through heuristics, game theory and argumentation. The most relevant approach that can enable sophisticated forms of interaction to support negotiation is argumentation. However, providing agents with appropriate conceptual models and related software architectures to fully automate argumentation and negotiation is still an unsolved research challenge.

To tackle this problem, we rely on a novel approach to the design of agent-based artificial societies, based on the notion of *artifact* for multi-agent systems (Omicini, Ricci and Viroli, 2008). Artifacts are used to model those abstractions in a multi-agent system environment that agents use to achieve individual and social goals. On the one hand, an artifact can serve the private purpose of an agent, acting, for example, as an external memory resource for placement and later retrieval of knowledge (e.g., a sort of agenda artifact). On the other hand, an artifact can be used as a social construct shared by multiple agents to mediate their interactions, for example, as a coordination medium for supporting an auction.

This work presents a conceptual framework based on agents and artifacts for negotiation dialogues, which uses argumentation between autonomous software agents to enable dialogue automation. Not only can agents generate new arguments themselves and assess the arguments they receive, but the system itself can assess proposed deals for acceptability to all the agents concerned. To enable this to happen, agents are given dialogue and argumentation artifacts by the infrastructure, artifacts which are used to calculate social notions such as argument acceptability, and to undertake assessments of proposed deals, in order to find a common agreement between all the participating agents. Moreover, all such artifacts can be either individual to each agent or be social, to more than one agent, because they provide different perspectives of the system evolution: the individual point of view is a collection of events and arguments produced by a singular agent (it is like the personal page of a social network); the social point of view is the sequence of events and arguments exchanged by all the agents in the society (it is like the home page of a social network). The main benefit from such an architecture is the possibility to exploit intelligent mediation services focused on individual or social information inside a dialogical process.

[1] We follow (Walton and Krabbe, 1995) in defining *negotiation dialogues* as dialogues over the division of some scarce resource.

The remainder of this paper is organized as follows. In Section 2 we provide the background knowledge for understanding the SANA framework, by describing in detail the elements of the referenced argumentation framework for SANA, and discussing the notion of artifact for multi-agent systems as defined in the A&A meta-model. In Section 3 we present the general architecture based on artifacts to enable the automatic form of negotiation. In Section 4 we discuss the negotiation dialogue introducing the SANAP protocol with the related denotational semantics. In Section 5 we present a case study where we apply our conceptual framework and the negotiation protocol, and we conclude the paper in Section 6 with a discussion of related and future work.

2 Background

2.1 SANA Argumentation Framework

In this section we introduce the notion of argumentation system that is the reference for our approach. Following (Prakken and Vreeswijk, 2002), we define an argumentation system as comprising a logical language, a set of arguments defined from this language, and a defeat relation defined over the set of arguments. The object language of our system is a first-order language, where Σ contains all well-formed formulae. The symbol $\vdash$ denotes classical inference (different modes, such as deduction, induction and abduction, will also be used), $\equiv$ denotes logical equivalence, and $\neg$ or non is used for logical negation. Inconsistent information is allowed in the knowledgebase Σ, using inference only with consistent subsets of that knowledgebase. The eventual inference conflicts are considered and managed at the argumentation level basing on the notion of argument and argument acceptability. An argument is a consistent subset of Σ together with the inference rules from that subset. The argument acceptability provides the criteria to collect arguments in acceptable sets.

Definition 2.1 (argument)**.** An *argument* is a triple $A = \langle B, I, C\rangle$ where $B = \{p_1, \ldots, p_n\} \subseteq \Sigma$ is a set of *beliefs*, $\vdash_I \in \{\vdash_d, \vdash_i, \vdash_a\}$ is the inference style (respectively, *deduction*, *induction*, or *abduction*), and $C = \{c_1, \ldots, c_n\} \subseteq \Sigma$ is a set of *conclusions*, such that:

1. B is consistent
2. $B \vdash_I C$
3. B is minimal, so no subset of B satisfying both 1 and 2 exists

An argument in classical logic is a sequence of inferences that leads to a conclusions and it is composed of three elements: beliefs, inference rules and conclusions. The types of formal inference that we consider for deduction, induction and abduction are shown in Table 1. Modus Ponens (MP) is a particular case of Multi-Modus Ponens (MMP) with only one premise. The inference process θ-subsumption derives a general rule R from specific beliefs B, but is not a legal inference in a strict sense.
MP is not complete in first order logic which means that it is not always possible to verify the validity of an argument. For example, a procedure for verification may not terminate when

Deductive Inference		Inductive Inference	
MP	$\frac{A \quad A \rightarrow B}{B}$	θ-su	$\frac{B}{R} \; where \; R\theta \subseteq B$
MT	$\frac{\neg A \quad B \rightarrow A}{\neg B}$	Abductive Inference	
MMP	$\frac{B_1,...B_n \quad (B_1,...B_n) \rightarrow C}{C}$	Ab	$\frac{B \quad A \rightarrow B}{A}$

Table 1: Deductive Inference: (MP) Modus Ponens, (MMP) Multi-Modus Ponens and (MT) Modus Tollens; Inductive and Abductive Inference: (θ-su) θ-subsumption, (Ab) Abductive

the conclusion is not provable from the premises. To deal with this problem, we introduce an operator (called $non/1$) for the negation of a conclusion. With this operator, an argument may be verified as the truth of non-conclusion, where non-conclusion belongs to the underlying knowledge base.

Following this, we provide a sequence of argument examples in first-order logic which exploit the different modes of inference. The arguments are represented as logic tuples through the predicate *argument* with the function *name* and other predicates such as *beliefs*, *infer* and *conclusions* to represent the triple $A = \langle B, I, C \rangle$.

Example 1. (Deductive Argument) The sentence *All men are mortal, Socrates is a man, Socrates is mortal* is an argument in deductive form which can be written in tuple form as follows:

$$\texttt{argument}(\texttt{name}, \texttt{beliefs}([\texttt{human}(\texttt{socrates})], [\texttt{clause}(\texttt{mortal}(\texttt{X}), [\texttt{human}(\texttt{X})])]),$$
$$\texttt{infer}(\texttt{MP}), \texttt{conclusions}([\texttt{mortal}(\texttt{socrates})])).$$

Example 2. (Argument from Modus Tollens) The sentence *All humans are mortal but Heraclitus is not mortal then Heraclitus is not human* is a argument form modus tollens inference which can be written in tuple form as follow:

$$\texttt{argument}(\texttt{name}, \texttt{beliefs}([\texttt{non}(\texttt{mortal}(\texttt{heraclitus}))], [\texttt{clause}(\texttt{mortal}(\texttt{X}),$$
$$[\texttt{human}(\texttt{X})])]), \texttt{infer}(\texttt{MT}), \texttt{conclusions}([\texttt{non}(\texttt{human}(\texttt{heraclitus}))])).$$

Example 3. (Inductive Argument) The sentence *All man are mortal* could be θ-subsumed from the sentences *Socrates is a man. Socrates is mortal*, which can be written in tuple form as follow:

$$\texttt{argument}(\texttt{name}, \texttt{beliefs}([\texttt{mortal}(\texttt{socrates}), \texttt{human}(\texttt{socrates})]),$$
$$\texttt{infer}(\texttt{Su}), \texttt{conclusions}([\texttt{clause}(\texttt{mortal}(\texttt{X}), [\texttt{human}(\texttt{X}))])])).$$

Example 4. (Abdutive Argument) The sentences *All humans are mortal, Parmenides is mortal, then Parmenides is a human*, comprise an argument from abductive inference which can be written in tuple form as follows:

$$\texttt{argument}(\texttt{name}, \texttt{beliefs}([\texttt{mortal}(\texttt{parmenide})], [\texttt{clause}(\texttt{mortal}(\texttt{X}),$$
$$[\texttt{human}(\texttt{X})])]), \texttt{infer}(\texttt{Ab}), \texttt{conclusions}([\texttt{human}(\texttt{parmenide})]).$$

The system allows defeat of deductive arguments from abductive and inductive arguments with unsound forms of inference at the same level. Abduction and induction rules of inference have

a non-monotonic behaviour: adding new premises means that the previous conclusions may turn out to false. Considering this possibility we exploit the acceptability semantics to manage the possible future conflict between new arguments based on new evidence and the previous arguments. In future work we may consider a hierarchy of inferences in the attack relation and a value to measure strength or weakness of an argument based upon the strength of the underlying reasons supporting that argument.
In respect to classical logical negation between two predicates, there are different types of attack or conflict between two arguments such as *undercut* and *rebuttal*, as defined in (Prakken and Vreeswijk, 2002). The notion of *undercut* means that an argument's conclusion conflicts with another argument's premise. The notion of *rebuttal* means that two arguments have conflicting conclusions.

Definition 2.2 (undercut)**.** Let $A_1 = \langle B_1, I_1, C_1 \rangle$ and $A_2 = \langle B_2, I_2, C_2 \rangle$ be two distinct arguments, A_1 is an *undercut* for A_2 iff $\exists h \in C_1$ such that $h \equiv \neg b_i$ where $b_i \in B_2$

Definition 2.3 (rebuttal)**.** Let $A_1 = \langle B_1, I_1, C_1 \rangle$ and $A_2 = \langle B_2, I_2, C_2 \rangle$ be two distinct arguments, A_1 is a *rebuttal* for A_2 iff $\exists h \in C_1$ such that $h \equiv \neg c_i$ where $c_i \in C_2$

Both *undercut* and *rebuttal* are binary relations of pairs of arguments; they will collectively be referred as *defeat* or attack relation.
From an algorithmic point of view, it is necessary to identify the contrasting predicate in order to decide between conflicting arguments. In order to identify the opposite predicate in our framework we introduce $non/1$ and $contrary/2$ operators respectively the first to identify the opposite predicate as $non(mortal(Socrates))$ is opposite to $mortal(Socrates)$ and the second to find specific relation of opposition like *faraway vs. near* $contrary(faraway, near)$.
The acceptability semantics of an argumentation system can be viewed as the set of principles to manage any inconsistency of information at argument level. Our semantics follows the definitions of acceptability and admissibility proposed by Dung (Dung, 1995). In particular, the main notions of conflict-free set, admissible set and preferred extension are provided.

Definition 2.4 (conflict-free set)**.** An argument set S is a *conflict free* set iff there exist no $A_i, A_j \in S$ such that A_i attacks A_j.

Definition 2.5 (collective defense)**.** An argument set S *defends collectively* all its elements if $\forall$ argument $B \notin S$ where B attacks $A \in S \quad \exists\ C \in S : C$ attacks B.

Definition 2.6 (admissible set)**.** An argument set S is a *admissible* set iff S is conflict free and S defends collectively all its elements.

Definition 2.7 (preferred extension)**.** An argument set S is a *preferred extension* iff S is a maximal set among the admissible set of A.

An argument is acceptable in the context of preferred semantics if an argument belongs in some/all preferred extensions (credulous/sceptical acceptance).

Definition 2.8 (credulous acceptability)**.** An argument A is *credulously acceptable* if A is contained in at least one preferred extension.

Definition 2.9 (sceptical acceptability)**.** An argument A is *sceptically acceptable* if A is contained in all preferred extensions.

The framework we describe below is neutral regarding the argumentation acceptability semantics.

2.2 Artifacts in the A&A Meta-Model

According to the A&A meta-model (Omicini et al., 2008), *agents* and *artifacts* are the fundamental building blocks for multi-agent systems (MAS). Agents play the role of the active and goal-oriented components that proactively undertake activities in order to achieve system goals. Artifacts are instead the passive, reactive entities that agents individually or cooperatively exploit during their activities. Artifacts make it possible to explicitly model MAS societies and environment, which then become first-class abstractions for MAS engineering (Ricci, Viroli and Omicini, 2008).

Taking human society as a metaphor, agents play the role of humans, while artifacts coincide with the objects and tools used by humans as either the means to support their work and achieve their goals, or the target of their activities. So, artifacts are *used* by agents: on the one side, they mediate the interaction between individual components and their environment (including the other components); on the other side, they embody the portion of the environment that can be designed and controlled to support agents' activities.

Unlike agents, artifacts are not meant to be autonomous or exhibit a pro-active behaviour, neither to have social capabilities. Among the main properties that are useful according to artifact purpose and nature (Omicini, Ricci and Viroli, 2006a), one could list: *(i) inspectability* and *controllability*, i.e. the capability of observing and controlling artifact structure, state and behaviour at runtime, and of supporting their on-line management, in terms of diagnosing, debugging, testing; *(ii) malleability* (or, *forgeability*), i.e. the capability of artifact function to be changed / adapted at runtime (on-the-fly) according to new requirements or unpredictable events occurring in the open environment, *(iii) linkability*, i.e. the capability of linking together distinct artifacts at runtime as a form of dynamic composition, as a means to scale up with complexity of the function to provide, and also to support dynamic reuse, *(iv) situatedness*, i.e. the property of being immersed in the MAS environment, and to be reactive to environment events and changes. It is worth remarking that most of these artifact features are not agent features: typically, agents are not inspectable, do not provide means for malleability, do not provide operations for their change, and do not compose with each other through operational links.

Artifacts can represent either the resources or objects that are directly the objective of agent activities, or the tools that are used as a medium to achieve such objectives. An example for the first case is a database or a knowledge repository in general, used to store and retrieve information. An example for the second case is given by a blackboard, used by agents as a tool to communicate and coordinate. In this paper, we focus on the latter notion of artifact, which is typically referred to as a *coordination artifact* (Omicini, Ricci, Viroli, Castelfranchi and Tummolini, 2004).

According to (Omicini, Ricci and Viroli, 2006b), coordination artifacts are artifacts designed to handle interaction between agents in a MAS. As such, a coordination artifact is an essential abstraction for building social activities, in that it is crucial both for enabling and mediating agent interaction, and for governing the social activities by ruling the space of agent interaction. Examples range from artifacts for concurrency management – such as semaphores, synchronisers, barriers, etc. –, to artifacts for communication management – such as blackboards, event services –, up to artifacts with articulated behaviours, such as workflow engines or auction engines. So, coordination artifacts generalise the common notion of coordination medium from the field of coordination models and languages (Papadopoulos and Arbab, 1998): simply put, coordination artifacts are the artifacts for MAS encapsulating the activity of MAS coordination—thus including, in principle, dialogue, negotiation and argumentation.
Since our focus here is showing how artifacts can be used along with argumentation for the support of negotiation in MAS, in the remainder of this paper we will use the general term of artifact while essentially referring to a refinement of the notion of coordination artifact in terms of individual and social artifacts, mainly exploited in Section 3 below.

3 SANA General Architecture

The SANA architecture, depicted in Figure 1, is based, first of all, on the distinction between individual and social artifacts. The Figure 1 shows the main parts of the architecture composed of agents, artifacts and their structural connection. Each individual artifact is connected with the corresponding social artifact.

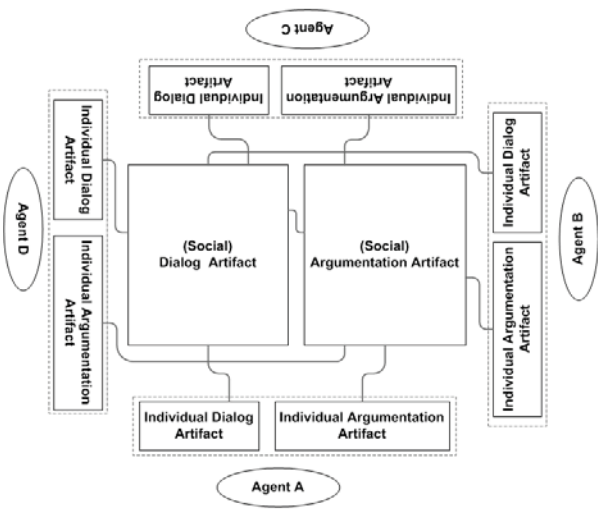

Figure 1: Overview of the System Architecture: the ellipses represent agents, the rectangles represent artifacts and the dashed line groups individual artifacts of a single agent.

According to the A&A meta-model (Omicini et al., 2008), individual artifacts in Figure 2 rule the

interaction of individual agents with the system, contain the laws and algorithms that govern the observable behaviour of individual agents, and store the observable portions of the individual data and information. So, the first essential component of our architecture is the Individual Dialogue Artifact (IDA). Every agent is associated with its own IDA, which provides the agent with the available operations that the agent can use during a dialogue, and also stores the history of the individual acts in the dialogues, thus providing an individual view over dialogues. In order to protect agent autonomy and privacy, while allowing the required information over their internal argumentation process to be available to the system's computation, a further individual artifact is provided: the Individual Argumentation artifact (IAA). The IAA contains the internal arguments of each agent, which can be useful during an argumentation process (by the SAA–see below), but that need to be kept secret / hidden from other agents.

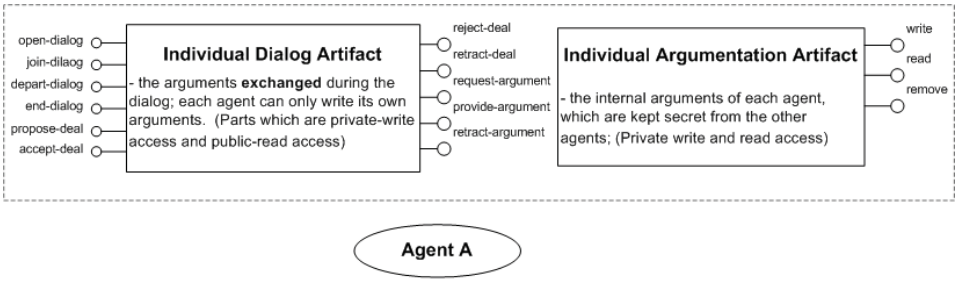

Figure 2: Individual Artifacts

On the other hand, social artifacts govern the interaction among agents within a MAS. First of all, the Dialog Artifact (DA) is a social artifact containing the rules and algorithms that record and govern the social aspects of dialogue. Accordingly, the protocol rules are contained here, along with the record of history of dialogues among agents, providing a global view over dialogues within a MAS. A social viewpoint over argumentation is instead contained in the Social Argument Artifact (SAA). SAA exploits public arguments from the IDA along with the private arguments made available by the IAA (see above), and builds social notions such as (social) acceptability, (social) conformance, etc., based on own algorithms such the ones in (Oliva, McBurney and Omicini, 2008).

3.1 Social Argumentation Artifact (SAA)

The notion of SAA comes from our previous work (Oliva, McBurney and Omicini, 2008) where we defined an artifact for argument computation called Co-Argumentation Artifact (CAA): in the general architecture defined in this paper, CAA corresponds to SAA here, and is then defined as follows.

Definition 3.1. A Social Argumentation Artifact (SAA) has only a public-read access and it is composed of an argumentation component (AC) and an argumentation store (AS) where:

- The Argumentation Store AS is a collection of arguments

- The Argumentation Component AC is a collection of algorithms that work over the collection of arguments AS.

The SAA is connected with IAA and IDA of each agent and it is unique during a dialogue session. It is the place where a group of agents could potentially find a mutually-acceptable proposal to directly resolve a negotiation dispute without having to undertake a dialogue. The agents have only a public-read access to this artifact.

Argumentation Component (AC)

The AC is a subsystem in charge of controlling a set of conflicting arguments, encapsulating well-known algorithms from the literature. Our current implementation of the AC uses preferred semantics, and its main functionalities are (i) to calculate the preferred extensions of a set of arguments, and (ii) to determine whether a new argument is valid and acceptable. In AC the argument structure and validity are considered at model level (looking inside arguments) while the argumentation semantics and arguments sets are considered at an upper level (using arguments as whole). The AC exploits a meta-reasoning approach to manage the argument set, which captures at the meta-level the properties between arguments; in particular, the meta programs should be able to calculate: (1) the relations of undercut and attack between arguments; (2) the conflict-free sets; and (3) the preferred extensions. The core of the AC is represented by the meta-interpreters that manage the arguments in order to find the conflict free sets, the admissible sets, and the preferred extensions.
The problem of a maximal conflict free set is already known in graph theory with the name of stable set or independent set: it is in the class of NP-hard problems. That is a result of worst-case analysis, while our objective is to have an algorithm that performs acceptably in average or typical cases. So, our idea is to build an algorithm that works incrementally, so as to avoid the complexity of an increasing amount of information. An incremental algorithm is also appropriate because we foresee a dynamic and distributed scenario where agents share their own arguments at different times. The system starts to re-calculate the admissible sets from the previous result.
In order to solve the conflict-free problem, we adopt a constraint-based approach with a standard backtracking strategy; for some code example and implementation detail we refer the interested reader to our previous work (Oliva, McBurney and Omicini, 2008).

Argument Store (AS)

The argument store provides a collection of socially-acceptable arguments—roughly speaking, it records and makes available to agents what is socially acceptable. Those arguments are either shared from public and private stores connected by construction by the SAA following a pre-fixed argumentation semantics, such as skeptical or credulous acceptance, or derived from the dialogue process as commitments. Such a collection is updated automatically every time a relevant modification of the connected artifacts occurs in term of argument modification.

3.2 Social Dialogue Artifact (SDA)

The Social Dialogue Artifact (SDA) introduced in (Oliva, Viroli, Omicini and McBurney, 2008) supports and mediates argumentative communication between agents. The SDA permits information, data and arguments, to be exchanged, and their public commitments to be recorded. Also, it implements the acceptability semantics of the dialogical process containing the rules that govern the evolution of the dialogue.
The SDA can support the dialogue by holding the information present in the connected artifact. The access to individual and social information may allow the SDA to elaborate over such stores while not revealing information private to individual agents.

Definition 3.2. A Social Dialogue Artifact is a triple SDA$= \langle DP, DS, IC \rangle$, where

- DP is a collection of specifications of dialogue protocols
- DS is a collection of utterances
- IC is a collection of specifications of interaction control (IC)

The SDA is connected with the individual and social artifact as IAA, IDA and SAA. A unique instance of the SDA is created for each dialogue session. The DP, DS and IC components are in turn defined in the following subsections.

Dialogue Protocols (DP)

The class DP is a collection of formal specifications of dialogue protocols, a possible realization is made using a labelled process algebra, as shown in (Oliva, Viroli, Omicini and McBurney, 2009). Protocols in DP may also be annotated with identifiers and with their properties, such as their termination complexity. When agents engage in dialogue using a protocol in the collection DP, they make utterances according to the permitted sequences defined by the protocol specification. Accordingly, the Dialogue Artifact is able to verify that utterances proposed by agents in a dialogue are valid under the protocol; the DA is also able to use the specification to suggest potential legal utterances to participating agents at each point in the dialogue.

Dialogue Store (DS)

For any particular collection of agents and any particular dialogue they undertake, the DS stores the sequence of locutions exchanged during the dialogue session. It provides a store for the dialogue as a whole with a declarative representation of the contents and a proper classification that follows the formal dialogue protocol syntax in use as the SANAP protocol in Section 4.

Interaction Control (IC)

The third component of the Social Dialogue Artifact, denoted as IC, is a collection of specifications for interaction control. A possible representation of the control rule, experimented in

(Oliva et al., 2009) is provided by the labelled transition system, modelling the progress over time of the agent interaction protocol. Three operators can be used to control the dialogue:

$$next^I(s) = \left\{ i : s \xrightarrow{i} s' \right\} \; next^S(s) = \left\{ s' : \exists i, s \xrightarrow{i} s' \right\} \; next^{IS} = \left\{ (i, s') : s \xrightarrow{i} s' \right\}$$

where s represents the state of the dialogue (by the sequence of uttered locutions), s' the new dialogue state and i the interaction (or current uttered locution). Operator $next^I(s)$ yields the next admissible interactions i from state s. Operator $next^S(s)$ yields the states reachable from s in one step. Operator $next^{IS}$ yields couples (i, s) instead.
The IC component realises the above three operators in order to identify which potential utterances are legal for any agent at any point in the dialogue. The basic primitives `in`, `rd`, `out` to manage arguments and facts in argument stores allow the IC to identify which constraints on the future course of dialogues are created by the existing commitments. For instance, the IC could permit only one utterance in a choice point, basing the decision on the state of the argument store. Also, it could work with an argument set over some advanced structures such as conflict-free sets and preferred extensions presented in Subsection 3.1 to determine for instance the acceptability of an argument.

3.3 Individual Argumentation Artifact (IAA)

The Individual Argumentation Artifact (IAA) is a private store associated to each agent, which records confidential information such as the agent's private valuations of some scarce resource (in the case of negotiation dialogues), or arguments based on privileged information (in the case of dialogues over beliefs).
The basic operations exploited by agents to store and retrieve arguments from and to the IAA are `read`, `write` and `remove` respectively to seek, add and retrieve arguments. Such private information is also accessible by social artifacts (SAA and SDA) exploiting the structural connection provided by the artifact linkability functionality. The SAA and SDA made `read` operations on the individual artifacts in order to collect all arguments and making possible some automatic elaboration for the construction of social notions such as (social) acceptability.

3.4 Individual Dialogue Artifact (IDA)

The Individual Dialogue Artifact (IDA) provides the basic operations that an agent can use to participate in an argumentative dialogical process and records the sequence of individual acts made by the agents during the dialogue. The IDA permits the eleven legal locutions presented below in the next section on the SANAP protocol. The denotational semantics of these locutions are also presented below.

4 SANAP Negotiation Protocol

In this section we present the formal syntax and denotational semantics for a multi-agent protocol for negotiation, called **SANAP**. We begin by presenting the ideas underlying the protocol informally, in order to motivate the syntax and semantics.

4.1 Motivation and structure

We consider an agent dialogue to be a sequence of utterances, also called *moves*, made by the agents participating in the dialogue. Each dialogue has at least two agents participating in it. For simplicity, we assume that agent utterances occur at different points of time in the dialogue, and that the system support infrastructure precludes simultaneous utterances.
We assume that the participants are seeking to allocate some scarce resource between themselves, as in the Negotiation Dialogues of (Walton and Krabbe, 1995). An *offer* or *proposed deal* in a negotiation represents a potential allocation of the resource, suggested either by one of the participants or by the support system infrastructure, and exchanged during the dialog. For instance, if the agents are negotiating for an increase in salary the offers will be the various amounts proposed. A negotiation dialogue ends in the following circumstances:

- When all the participants reach agreement on a proposed deal.
- When the second-last participant leaves the dialogue (i.e, when there is only one participant remaining in the dialogue).

Otherwise, the dialogue does not end, and continues. For the purposes of this paper, we assume there are two main types of negotiation dialogue — Competitive and Collaborative.

Competitive Negotiation involves selfish participants, with each participant seeking only to maximize its own share of the resource. The dialogue between the participants comprises a sequence of arguments for (pro) or against (con) a particular proposal (or offer) to allocate the resources under discussion. The winner of this form of negotiation is the agent with the most powerful arguments, defined according to an agreed argumentation semantics of acceptable arguments, for example, a preferred extension of all the exchanged arguments. The system infrastructure supports the exchange and storage of proposals, and the exchange, identification, resolution and storage of arguments pro and con these proposals. The infrastructure also guides the participants by constraining the arguments in a sequence of attacking arguments and by evaluating the winning arguments.

Collaborative Negotiation involves participants jointly sharing their private arguments with the goal of finding the allocation of resources best supported by the joint collection of arguments. The SANA system infrastructure allows participants to combine their private arguments without other agents being able to see their individual arguments; thus a level of privacy is maintained, even with the joint pooling of arguments. After the arguments have been stored, the infrastructure collects the arguments from each private space and then proposes a set of agreed arguments. From these agreed arguments, an acceptable offer may be inferred, being a consequence of the agreed arguments.

Generally arguments are exchanged during the dialogue as justification for a specified proposal, or for certain belief, or an agent choice or preference. An argument supports and justifies an offer only if the offer is in the conclusion of that argument.
For Competitive Negotiations, it is possible to structure dialogues between two or more agents as a sequence of arguments and counter-arguments about a proposed offer:

- An agent makes a proposed offer, and this is followed by arguments for and against it by different agents.
- Discussion over a proposed offer ends when no further arguments can be presented by any participant, for or against the proposed offer.
- If the exchange of arguments has ended (ie, if no participant is able or willing to present any further arguments), and if the final argument presented is in support of the proposed offer, then we would expect all rational participants to accept the offer. In that case, following their explicit acceptance, the dialogue would end with that offer as the deal between the participants. The participants would normally be bound to implement the deal contained in the accepted offer, in the world outside the dialogue.
- If the final argument presented is not in support of the proposed offer, then we would expect a rational proposer to retract the offer. In that case, the dialogue is open to any agent to make further proposed offers.

In contrast to this process, for Collaborative Negotiations it is possible to avoid a dialogical process, by exploiting the SANA architecture, in particular, the fact that the arguments of all participants are stored in the private IAA.

- Agents store whatever arguments they wish about a proposed offer in their private IAA.
- When no further argument is presented by the participants, the system infrastructure, acting as a neutral mediator, collects all the arguments from each private IAA and undertakes an automated resolution to determine an appropriate subset of acceptable arguments according to an agreed argumentation semantics, for example, a preferred extension. The SANA framework is intended to be neutral regarding the argumentation framework acceptability semantics used by the mediator to decide acceptable arguments: the participants may use whichever semantics they agree to use.
- The system infrastructure then undertakes an automated evaluation of the arguments in that set of acceptable arguments to infer the set of possible offers which are the conclusions of arguments in the acceptable set.
- We would then expect rational participants to accept one of the offers in that set of possible offers determined by the system infrastructure.

Real negotiation dialogues may combine elements of both the Competitive and the Collaborative Negotiations we have identified here. In addition, for implementation of this support framework in an open environment, it would probably be necessary to undertake conformance checking to ensure that the arguments contained in agent utterances are legal, relevant, and do not conflict with prior commitments of the participants. We leave that issue for future work.

4.2 SANA Negotiation Protocol: Syntax

We now present a formal syntax and combination rules for the legal utterances that agents may make in the course of a dialogue under the SANA protocol. Our syntax for utterances will be:

illocution(a_i, ϕ) or *illocution*(a_i, a_j, ϕ)

where *illocution* is an illocution, a_i is an identifier for the agent making the utterance (the speaker), $a_j \neq a_i$ denotes an agent a_j at whom the utterance by a_i is directed, and ϕ is the content of the utterance. For simplicity we assume all utterances are made to the entire group involved in the dialogue. For the content of the utterance, any agreed formal language may be used. We will assume the content layer is represented in a propositional language, with lower-case Greek letters as propositions and upper-case Greek letters as topics of discussion. We denote the set of these well-formed content formulae, closed under the usual connectives, as $\mathcal{C}$. These propositions may represent objectively-verifiable statements about the world, or internal preferences, or intentions, or commitments, etc. Because we wish to use the protocol to exchange justifications for claims, some utterances will also have content comprising arguments, in the format defined in Section 2 above. We denote the set of these arguments as $\mathcal{A}$, and individual arguments by A, B, etc. Finally, we assume that time is discrete and may be represented by the natural numbers, and that precisely one utterance occurs on each time-step. For simplicity, we do not include a time stamp in the syntax.

We now define the legal locutions of the multi-agent negotiation protocol:

L1: open-dialog(a_i, a_j, Φ**):** A speaker a_i expresses a desire to initiate a dialogue with agent a_j concerning topic $\Phi \in \mathcal{C}$. For any given topic Φ, we allow only one **open-dialog(.)** utterance to be made by an agent a_i while that initial dialog remains open. (In other words, an agent cannot initiate more than one simultaneous dialog on a given topic.)

L2: join-dialog(a_j, a_i, Φ**):** A speaker a_j expresses a willingness to join a dialogue with agent a_i concerning topic $\Phi \in \mathcal{C}$

L3: depart-dialog(a_i, a_j, Φ**):** A speaker a_i expresses an intention to depart the dialogue with agent a_j concerning topic $\Phi \in \mathcal{C}$ which was initiated in an earlier *open-dialog(.)* utterance.

L4: end-dialog(a_j, a_i, Φ**):** A speaker a_i expresses the cessation of the dialogue with agent a_j concerning topic $\Phi \in \mathcal{C}$ which was initiated in an earlier *open-dialog(.)* utterance.

L5: propose-deal(a_i, a_j, ϕ**):** A speaker a_i proposes to agent a_i a deal represented by formula $\phi \in \mathcal{C}$. Typically, ϕ will represent a division of a resource between the two agents.

L6: accept-deal(a_j, a_i, ϕ**):** A speaker a_j expresses a willingness to accept a deal ϕ proposed by agent a_i in an earlier *propose-deal(.)* utterance.

L7: reject-deal(a_j, a_i, ϕ**):** A speaker a_j expresses rejection of a deal ϕ proposed by agent a_i in an earlier *propose-deal(.)* utterance.

L8: retract-deal(a_i, a_j, ϕ**):** A speaker a_i retracts an earlier utterance of a proposed deal ϕ proposed to agent a_i.

L9: request-argument(a_m, a_k, ϕ**):** A speaker a_m requests an argument from agent a_k in justification for statement ϕ.

L10: provide-argument(a_k, a_m, A**):** A speaker a_k provides an argument $A \in \mathcal{A}$ to agent a_m.

L11: retract-argument(a_k, a_m, A): A speaker a_k retracts an argument $A \in \mathcal{A}$ previously provided to agent a_m.

Locutions L1 through L4 control the start and end of the dialogue, and the entry and departure of its participants. Locutions L5 through L8 manage the presentation of and response to proposals for deals. Locutions L9 through L11 manage argument about proposals already made via Locution L3, prior to their response via Locutions L4, L5 and L6. As with other agent interaction protocols (e.g., (McBurney and Parsons, 2002)), the protocol syntax includes the definition of several combination rules, which control the order in which particular locutions may be uttered.

CR1: A dialogue can only begin with an utterance of **open-dialog(a_i, a_j, Φ)**.

CR2: An utterance of **open-dialog(a_i, a_j, Φ)** must be followed by an utterance of **join-dialog(a_j, a_i, Φ)** for a dialogue to open.

CR3: The utterance **propose-deal(a_k, a_m, ϕ)** may be made at any time by any agent a_k, following utterances of **join-dialog(a_k, a_i, Φ)** and **join-dialog(a_m, a_i, Φ)**.

CR4: The utterances **accept-deal(a_m, a_k, ϕ)** and **reject-deal(a_m, a_k, ϕ)** may be made at any time by agent a_m following the utterance of **propose-deal(a_k, a_m, ϕ)**.

CR5: The utterance **retract-deal(a_k, a_m, ϕ)** may be made at any time by agent a_k following the earlier utterance of **propose-deal(a_k, a_m, ϕ)**.

CR6: The utterance **depart-dialog(a_k, a_m, ϕ)** may be made at any time by any agent following an utterance of **join-dialog(a_j, a_i, Φ)**.

CR7: The utterance **end-dialog(a_i, a_j, ϕ)** may be made at any time by any agent a_i which had previously uttered **open-dialog(a_i, a_j, Φ)**.

CR8: An utterance of **request-argument(a_m, a_k, ϕ)** can only be uttered by agent a_m after a prior utterance by agent a_k of **propose-deal(a_k, a_m, ϕ)**.

CR9: An utterance of **provide-argument(a_k, a_m, A)** can only be uttered by agent a_k after a prior utterance by agent a_m of **request-argument(a_m, a_k, ϕ)**, and only when ϕ is the third element of the triple denoted by argument $A \in \mathcal{A}$.

CR10: An utterance of **retract-argument(a_k, a_m, A)** can only be uttered by agent a_k after a prior utterance by agent a_k of **provide-argument(a_k, a_m, A)**.

It is worth noting some principles which these rules instantiate. The **SANA** framework requires unanimous consent of the participants for a deal to be reached, and this includes agreement from whichever participant originally proposed the deal. Thus, a deal cannot be achieved if a proposer retracts the proposed deal after first proposing it. Retraction may occur at any time, including after one or more of the other participants have accepted the proposed deal. At this point, each of these other participants still has themself the possibility to propose the

same deal. In the case of more than two participants, the dialogue may continue even after the initiator has departed the dialogue. If unanimous agreement is not possible, then no deal is achieved; because the participants are autonomous, negotiation between them always risks such non-agreement. In the case where no deal is achieved, participants may maintain their presence in the dialogue as long as they individually wish, for example, if awaiting new information or a change of preferences by some participant. Because the participants are autonomous, they cannot be forced to remain in the dialogue against their will, but also nor can they be forced to leave it.

Although we have assumed unanimous agreement is required for a deal, other models of multi-party decision-making may be appropriate for particular application domains, eg, agreement by a majority, or a plurality, or decision by a dictator, etc. Such alternatives may be considered in future development of the framework.

Locutions L1 through L8, together with Combination Rules CR1 through CR7 specify a basic multi-agent negotiation protocol, which allows the participants to present and respond to potential deals, but which does not permit argument over these proposed deals. The additional locutions (L9 through L11) and combination rules (CR8 through CR10) further enable the participants to engage in argument about these proposed deals. We could therefore see these additional locutions and rules (L9 through L11, and CR8 through CR10) as providing an argumentation functionality to the protocol, just as the locutions and rules of the *Fatio Protocol* proposed in (McBurney and Parsons, 2005b) provide an argumentation functionality to the FIPA Agent Communications Language, *FIPA ACL* (FIPA, 2002).

4.3 SANA Negotiation Protocol: Denotational Semantics

There are many reasons for articulating a semantics for an agent interaction protocol, including to ensure that all participating agents and their software developers share a common understanding of the protocol syntax, and to facilitate its implementation as software.[2] Recently, several protocols have been given a denotational semantics which maps dialogue utterances to actions on a virtual tuple space or similar blackboard structure (Gelernter and Carriero, 1992; Omicini and Denti, 2001), for instance, (Doutre, McBurney and Wooldridge, 2005; McBurney and Parsons, 2005a; McBurney and Parsons, 2007). Experience with software implementations of these protocols indicates that such denotational semantics can be readily and correctly implemented, e.g., (Doutre, McBurney, Wooldridge and Barden, 2005).

We now present a denotational semantics for the agent negotiation protocol with the syntax defined above. We present the semantics only in outline form, since this is sufficient to understand the underlying principles. Our semantics maps an agent dialogue conducted according to the protocol to a partitioned tuple space, and maps the legal utterances in the dialogue to specified tuple-placement events in that space, as follows:

1. Each new dialogue is assumed to create a new, partitioned tuple space. This space is created when two agents utter the linked pair of locutions **open-dialog**, followed by **join-dialog**. If there are n agents in the dialog, then there are $3n + 3$ sub-spaces in the tuple

[2] The various different functions of ACL semantics are discussed in (McBurney and Parsons, 2009).

spaces, comprised as follows:

- For each agent a_i, there is a private store P_i for the agent's internal arguments and proposals. Only agent a_i may write to or delete from this store, and only the agent itself and the mediator may read this store. Deletion from this store may take place at any time for any reason. In terms of the system architecture defined in Section 3 above, each private store P_i may be viewed as part of the Individual Argumentation Artifact(IAA).
- For each agent a_i, there is a public argument store U_i for the agent's arguments presented in the dialog. Only agent a_i may write to or delete from this store, but any agent and the mediator may read it. Deletion from this store may only be effected by the utterance of an appropriate **retract-argument** locution and only according to the combination rules of the dialog. Each public argument store U_i may be viewed as part of the Individual Dialogue Artifact(IDA).
- For each agent a_i, there is a public deal store D_i for the agent's proposals presented in the dialog. Only agent a_i may write to or delete from this store, but any agent and the mediator may read it. Deletion from this store may only be effected by the utterance of an appropriate **retract-deal** locution and only according to the combination rules of the dialog. Each public deal store D_i may also be viewed as part of the Individual Dialogue Artifact(IDA).
- There is a sub-space for accepted deals, denoted AD. Any agent and the mediator may read from this space. Any agent (but not the mediator) may write to this space, but only by copying a proposed deal stored in one of the public deal stores D_i. No agent or the mediator may delete the contents from this space. The space for accepted deals AD may be viewed as part of the Social Dialogue Artifact(SDA).
- There is a sub-space for rejected deals, denoted RD. Any agent and the mediator may read from this space. Any agent (but not the mediator) may write to this space, but only by copying a proposed deal stored in one of the public deal stores D_i. No agent or the mediator may delete the contents from this space. The space for rejected deals RD may also be viewed as part of the Social Dialogue Artifact(SDA).
- There is a sub-space for inferred deals, denoted ID. Any agent and the mediator may read from this space. Only the mediator may write to or delete from this space. The space for inferred deals ID may be viewed as part of the Social Argumentation Artifact(SAA).

2. Whenever agent a_i utters **propose-deal**(a_i, a_j, ϕ) a tuple labeled (a_i, a_j, ϕ) is placed within the public deal store D_i of agent a_i.

3. Whenever agent a_j utters **accept-deal**(a_j, a_i, ϕ) a tuple labeled (a_i, a_j, ϕ) is placed within the accepted deals sub-space, AD.

4. Whenever agent a_j utters **reject-deal**(a_j, a_i, ϕ) a tuple labeled (a_i, a_j, ϕ) is placed within the rejected deals sub-space, RD.

5. Whenever the mediator agent infers from analysis of the private argument spaces of agents a_i and a_j that a possible deal ϕ may be acceptable to both agents, a tuple labeled (a_i, a_j, ϕ) is placed within the sub-space for inferred deals, ID.

6. Whenever agent a_i utters **retract-deal**(a_i, a_j, ϕ) the tuple labeled (a_i, a_j, ϕ) is removed from the public deal store D_i of agent a_i.

7. Whenever agent a_i utters **provide-argument(**a_i, a_j, A**)** a tuple labeled (a_i, a_j, A) is placed within the public argument store U_i of agent a_i.

8. Whenever agent a_i utters **retract-argument(**a_i, a_j, A**)** the tuple labeled (a_i, a_j, A) is removed from the public argument store U_i of agent a_i.

9. Once an agent a_i utters the locution **depart-dialog**, no further tuples are added to that agents private or public sub-spaces, P_i, U_i and D_i.

10. Once an agent utters the locution **end-dialog**, no further placement or movement of tuples takes place in the tuple space associated to the dialog.

From this outline, it is straightforward to articulate this protocol semantics formally, as undertaken previously for similar denotational semantics of agent protocols in (Doutre, McBurney, Wooldridge and Barden, 2005; McBurney and Parsons, 2007). It would also be straightforward to interpret this tuple-space semantics using category theory, as in (McBurney and Parsons, 2005a; McBurney and Parsons, 2007). Although technically not part of the semantic mapping between the formal syntax and the partitioned tuple space, dialogue participants may also agree that deals represented by tuples in the accepted deal store, AD, create binding commitments on the participants in the world outside the dialogue (either the physical world and/or other parts of the online world).

5 Case Study of Negotiation

In the following we propose an example application of the SANA framework for salary negotiations between a worker Alpha who would like to change jobs and a new company Beta, whom Alpha would like to work for. To each agent is associated a private IAA and IDA and two social artifact SAA and SDA common for both. The agents exploit the legal locution provided by IDA to communicate and the SAA to reach an agreement.
Alpha is attracted to the company Beta for the better chance of career. Beta is interested of Alpha for his competences. The starting request of Alpha is a salary of 1200 euro per month and Beta can offer a maximal salary of 900 euros per month. This situation seems to have no solution unless Alpha reduces his demand or Beta changes its salary policy. The situation could be change if the participants start to exchange arguments in support of their positions (competitive negotiation) and arguments that express their underlying interests (collaborative negotiation).

Competitive Negotiation

The two agents start a negotiation dialogue exploiting the legal locution provided by the IDA following the protocol described in the previous section. The worker Alpha would like to change work and he tries to negotiate his salary with a new company Beta.

```
open-dialog(worker(alpha),company(beta),salary)
join-dialog(company(beta),worker(alpha),salary)
```

The worker request 1200 euro per month. The company in response offers 900 euro per month.

```
propose-deal(worker(alpha),company(beta),salary(1200))
propose-deal(company(beta),worker(alpha),salary(900))
```

The competitive negotiation with arguments tries to find a zero cost solution where the “most powerful wins”. From this situation the negotiation continues with the concession from the weaker participant. The worker Alpha makes a new request, asserting that the minimum national legal salary is 1000 per month.

```
propose-deal(worker(alpha),company(beta),salary(1000))
request-argument(company(beta),worker(alpha),salary(1000))
provide-argument(worker,company(beta),argument([position(employee)
                 ,contract(national,employee,1000),
                 [salary(X):-position(N),contract(national,N,X)],
                 MP,salary(1000)).
```

The worker’s argument is very strong and the company Beta does not have a contrary argument. The winning argument will be stored also in the public SAA. Beta has two possible moves to accept the deal with the new offer augmenting the salary proposal to 1000 euros or to retract its offer and leave the dialog.

```
accept-deal(company(beta),worker(alpha),salary(1000))
retract-deal(company(beta),worker(alpha),salary(900))
```

Neither of these two responses may satisfy the larger goals of the participants: the worker wants to get a new job at a reasonable salary, and the company wants to employ a satisfied worker at an affordable salary.

Collaborative Negotiation

Continuing the previous example the two participants Alpha and Beta want to collaborate to find a common solution helpful for both. The key point of the collaboration is the sharing of private arguments of each agent that can be done automatically in SANA framework storing private information in the IAA. In this case the SAA receives that information, calculates the social agreement, and publishes the acceptable sets with new offers composed by the union of arguments conclusions.

In our example each agent stores the arguments in own IAA.The worker argues privately that the previous monthly salary was 1000 euros and the company was close to his home. These arguments are stored in the worker's IAA:

```
argument(argWsalary,[company(near),evaluate(near,1000),
         salary(X):-company(Distance),evaluate(Distance,X)],
         mp,salary(1000))
argument(argWprevious,[company(faraway),evaluate(faraway,1200),
        salary(X):-company(Distance),
        evaluate(Distance,X)],mp,salary(1200))
argument(argWfaraway,[myhome(roma)],_,company(faraway))
```

The company argues privately that it does not augment the monthly salary more than 1000 euros because the other new employees keep the same amount (argument for analogy). It argues too that it could pay the travel cost of the worker with a salary of 1000 euros. These arguments are stored in the company's IAA:

```
argument(argCall,[employee(newWorker,1000),
         salary(X):-employee(N,X)],
         mp,salary(1000))
argument(argCrefund,[company(faraway),salary(1000),evaluate(faraway,refund(yes))
         refund(X):-company(D),salary(S),evaluate(D,S,refund(X))],
         mp,refund(yes))
```

At the end of the presentation process the SAA collects all the arguments stored in the private IAAs by worker and company. A simplified version of the resulting arguments collection is provided as follow:

Worker - Alpha

```
argument(argWsalary,[company(near)],mp,[salary(1000)]).
argument(argWprevious,[company(faraway)],mp,[salary(1200)]).
argument(argWfaraway,[myhome(roma)],mp,[company(faraway)]).
argument(argWnational,[contract(national)],_,[salary(1000)]).
```

Company - Beta

```
argument(argCall,[employee(newWorker,1000)],mp,[salary(1000)]).
argument(argCrefund,[company(faraway),salary(1000)],_,[refund(yes)]).
argument(argCproposal,[contract(learner)],_,[salary(900)]).
```

After that the SAA undertakes an automatic reasoning process to determine the subsets of acceptable arguments as preferred or grounded extensions. The results of the computation is presented by the SAA with a collection of logic tuples as `conflictfreeset`, `admissibleset`, `preferredextension` and `groundedextension` containing lists of arguments names that determine the sets. In this example the maximal conflict free sets and the two extensions coincide.

```
conflicfreeset([argWprevious,argWfaraway],[argCproposal,argWfaraway],
           [argWnational,argWsalary,argWfaraway,argCall,argCrefund])
admissibleset([argWprevious,argWfaraway],[argCproposal,argWfaraway],
           [argWnational,argWsalary,argWfaraway,argCall,argCrefund]])
preferredexstension(
           [argCproposal,argWfaraway],[argWprevious, argWfaraway],
           [argWnational,argWsalary,argWfaraway,argCall,argCrefund])
```

In a negotiation dialogue an agreement is found when a proposal is acceptable to all the agents participants of the dialogue. Following preferred semantics an acceptable argument should belong at least one preferred extension. In the case that the preferred extension is empty this means that there are no acceptable proposals for the negotiating agents. In the other case it is possible to find an agreement in the preferred extensions composed of arguments from all the participant agents in the SAA. In our example a possible agreement is represented by the SAA with a new deal composed of all the conclusions belong to the preferred extension with arguments from both agents and that involve the salary.

```
deal([salary(1000),refund(yes)])
```

The agents are leaved to decided to accept or not the proposed solution. In this case both agents send a locution to accept the new deal.

```
accept-deal(worker(alpha),company(beta),[salary(1000),refund(yes)])
accept-deal(company(beta),worker(alpha),[salary(1000),refund(yes)])
```

The agreement reached meets the requirement of both agents: improving the contract condition of the worker Alpha and providing a motivated and satisfied employee to company Beta.

6 Related Work and Conclusions

In this paper, we have presented a novel framework, called **SANA**, for argumentation-based negotiation between multiple autonomous agents, in which the participants may collaborate to a greater or lesser extent, and in which an independent mediator agent or artifact is present. The framework includes an interaction protocol, **SANAP**, through which participants may share proposals for dividing some scarce resource, along with arguments for and against these proposals. Our framework and protocol enable the participants either to share all their arguments with one another, in a collaborative fashion, or, alternatively, only to reveal their arguments as and when they individually judge to be necessary to support their own proposals or to counter the proposals of others, competitively. In the case of collaborative sharing of arguments, the SANA framework includes an intelligent artifact which acts as mediator to the negotiation, seeking to infer proposals from the shared pool of arguments using an agreed dialectical argumentation theory. In the case of competitive negotiation, winning proposals are those whose supporting arguments defeat the counter arguments, again according to an agreed dialectical argumentation theory. The combination of an intelligent mediator artifact in a framework for

multi-agent negotiation which incorporate formal dialectical argumentation makes our framework novel, as far as we are aware.

There has been considerable interest in recent years in frameworks for computational negotiation between autonomous software agents, as a part of a wider interest in computational argumentation (Rahwan and Simari, 2009; Reed and Norman, 2004). For a review of research on argumentation-based negotiation, see the paper by Rahwan *et al.* (Rahwan, Ramchurn, Jennings, McBurney, Parsons and Sonenberg, 2003). Our work differs from earlier work in various respects. The very early work of Sycara (Sycara, 1990), although focused on modeling the specific domain of labour-company negotiations, shares with our framework a central mediator. However, that framework does not incorporate formal dialectical argumentation, which is not surprising in view of its publication before the seminal work of Dung (Dung, 1995). In addition, Sycara's framework provides less autonomy to the participants and more power to the mediator than does our framework. The later work of Parsons *et al.* (Parsons, Sierra and Jennings, 1998) models the agents participating in a negotiation with Belief-Desire-Intention (BDI) logics, in a framework which also does not incorporate formal dialectical argumentation. In addition, there is no mediator role in that structure. Similar comments apply to the multi-agent negotiation frameworks and protocols presented in (Amgoud, Parsons and Maudet, 2000; Jennings, Faratin, Lomuscio, Parsons, Wooldridge and Sierra, 2001; McBurney, Eijk, Parsons and Amgoud, 2003).

Other recent work in multi-agent negotiation has had different foci from our work here. For example, the research of Karunatillake and colleagues (Karunatillake, 2006; Karunatillake, Jennings, Rahwan and McBurney, 2009) has sought to identify the circumstance under which agent negotiators would benefit from using argumentation, as evidenced through computational simulation studies, while that of Amgoud and colleagues (Amgoud, Dimopoulos and Moraitis, 2008) has analyzed the outcomes achievable in multi-agent negotiations using an argumentation-based protocol. Other work, such as that of Rovatsos *et al.* (Rovatsos, Rahwan, Fischer and Weiss, 2006) and Rahwan *et al.* (Rahwan, Sonenberg, Jennings and McBurney, 2007), has considered the strategies appropriate to agents participating in negotiation, and the formal modeling of these. In addition, recent work by Rahwan and colleagues, e.g. (Rahwan, Larson and Tohme, 2009), has looked at the design of argumentation protocols to encourage or discourage particular participant strategies. We have not yet considered strategies for the participants in the SANA framework.

The use of artificial mediators has been a component of other research on multi-party argumentation not dealing specifically with agent negotiation, particularly in systems designed for public deliberation, as in (Gordon and Karacapilidis, 1997; Gordon and Richter, 2002; Rehg, McBurney and Parsons, 2005). Although not intelligent and not a mediator, the use of a central blackboard or store for sharing proposals in multi-agent negotiations can be found in the framework of Bratu *et al.* (Bratu, Andreoli, Boissier and Castellani, 2002), and in various recent papers on the semantics of multi-agent interaction protocols; see (McBurney and Parsons, 2009) for a review. This concept of a central blackboard can be traced back to both the *Commitment Stores* of Hamblin's dialogue game theory (Hamblin, 1970) and to tuple spaces and the associated *Linda* co-ordination language (Gelernter, 1985; Gelernter and Carriero, 1992).

In future work, we plan to consider the broad issue of conformance checking of utterances, ensuring for example that the arguments contained in agent utterances are legal, relevant, and not in conflict with the prior commitments of the speaker. As noted in (Rahwan et al., 2003) this topic has been somewhat neglected in the study of agent interaction protocols. Another possible topic for future investigation is the generation, selection and assessment of strategies for participants in agent negotiations, particularly at run-time. An interesting question under the heading of participant strategies is that of the circumstances under which agents should prefer either collaborative or competitive interactions using the SANA framework; in other words, when is it preferable for an agent to share all its arguments with other agents with whom it is in negotiation? The answers to such questions will of course depend on the nature of the other participants in the negotiation, and thus modeling of opponents, as in (Oren and Norman, 2010), will play a part in such analyses.

Finally, it may be valuable to briefly speak of potential applications of the **SANA** framework described here. Argumentation in general has a long history of application in medicine, for example, initially for automated reasoning under uncertainty; see (Carbogim, Robertson and Lee, 2000) for a review of that early work. More recent applications in medicine have included automated negotiation using argumentation for the matching of organ donors and recipients, for example (Tolchinsky, Atkinson, McBurney, Modgil and Cortes, 2007), using a framework called Proclaim (Tolchinsky, Modgil, Cortes and Sanchez-Marre, 2006). The same framework has been used for applications of automated negotiation with argumentation to resource allocation domains, such as resolving competing uses of water resources (Aulines, Tolchinsky, Turon, Poch and Cortes, 2007; Tolchinsky, Aulines, Cortes and Poch, 2009). Although Proclaim also includes a mediator agent to arbitrate between claims, the underlying arguments of this framework are past cases; in other words, the framework combines argumentation theory with case-based reasoning. In the SANA framework, by contrast, the underlying arguments are not necessarily past cases, but abstract arguments, and the system components (including the mediator) are specified more formally and more generally than in Proclaim. These uses of Proclaim show the potential applications of automated negotiation using argumentation, and in future work it would be interesting to apply the SANA framework to application domains such as these.

Acknowledgments

We are grateful to the European Commission's Information Society Technologies (IST) Programme for financial support through *Project ASPIC: Argumentation Service Platform with Integrated Components* (Project FP6-IST-002307). Earlier and preliminary versions of some of these ideas were presented at recent ArgMAS workshops (Oliva, McBurney and Omicini, 2007; Oliva, Viroli, Omicini and McBurney, 2008).

References

Amgoud, L., Dimopoulos, Y. and Moraitis, P. 2008. *Argumentation in Multi-Agent Systems*, Lecture Notes in Artificial Intelligence 4946, Springer, Berlin, Germany, chapter A general

framework for argumentation-based negotiation, pp. 1–17.

Amgoud, L., Parsons, S. and Maudet, N. 2000. Arguments, dialogue, and negotiation, *in* W. Horn (ed.), *Proceedings of the Fourteenth European Conference on Artificial Intelligence (ECAI 2000)*, IOS Press, Berlin, Germany, pp. 338–342.

Aulines, M., Tolchinsky, P., Turon, C., Poch, M. and Cortes, U. 2007. Is my spill environmentally safe? towards an integrated management of wastewater in a river basin using agents that can argue, *The 7th International IWA Symposium on Systems Analysis and Integrated Assessment in Water Management*.

Bratu, M., Andreoli, J. M., Boissier, O. and Castellani, S. 2002. A software infrastructure for negotiation within inter-organisational alliances, *in* J. A. Padget, O. Shehory, D. C. Parkes, N. M. Sadeh and W. E. Walsh (eds), *Agent-Mediated Electronic Commerce IV (AMEC-IV): Designing Mechanisms and Systems*, Lecture Notes in Artificial Intelligence 2531, Springer, Berlin, Germany, pp. 161–179.

Carbogim, D. V., Robertson, D. S. and Lee, J. R. 2000. Argument-based applications to knowledge engineering, *Knowledge Engineering Review* **15**(2): 119–149.

Doutre, S., McBurney, P. and Wooldridge, M. 2005. Law-governed Linda as a semantics for agent interaction protocols, *in* F. Dignum, V. Dignum, S. Koenig, S. Kraus, M. P. Singh and M. Wooldridge (eds), *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2005), Utrecht, The Netherlands*, ACM Press, New York City, NY, USA, pp. 1257–1258.

Doutre, S., McBurney, P., Wooldridge, M. and Barden, W. 2005. Information-seeking agent dialogs with permissions and arguments, *Technical Report ULCS-05-010*, Department of Computer Science, University of Liverpool, Liverpool, UK.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *Artificial Intelligence* **77**(2): 321–358.

FIPA 2002. Communicative Act Library Specification, *Standard SC00037J*, Foundation for Intelligent Physical Agents.

Gelernter, D. 1985. Generative communication in Linda, *ACM Transactions on Programming Languages and Systems* **7**(1): 80–112.

Gelernter, D. and Carriero, N. 1992. Coordination languages and their significance, *Communications of the ACM* **35**(2): 97–107.

Gordon, T. F. and Karacapilidis, N. 1997. The Zeno argumentation framework, *Proceedings of the Sixth International Conference on AI and Law*, ACM Press, New York, NY, USA, pp. 10–18.

Gordon, T. F. and Richter, G. 2002. Discourse support systems for deliberative democracy, *in* R. Traunmüller and K. Lenk (eds), *EGOV 2002*, LNCS 2456, Springer, Berlin, Germany, pp. 238–255.

Hamblin, C. L. 1970. *Fallacies*, Methuen, London, UK.

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

Karunatillake, N. C. 2006. *Argumentation-Based Negotiation in a Social Context*, Phd, School of Electronics and Computer Science, University of Southampton, Southampton, UK.

Karunatillake, N. C., Jennings, N. R., Rahwan, I. and McBurney, P. 2009. Dialogue games that agents play within a society, *Artificial Intelligence* **173**(9–10): 935–981.

McBurney, P., Eijk, R., Parsons, S. and Amgoud, L. 2003. A dialogue-game protocol for agent purchase negotiations, *Journal of Autonomous Agents and Multi-Agent Systems* **7**(3): 235–273.

McBurney, P. and Parsons, S. 2002. Games that agents play: A formal framework for dialogues between autonomous agents, *Journal of Logic, Language and Information* **11**(3): 315–334.

McBurney, P. and Parsons, S. 2005a. A denotational semantics for deliberation dialogues, *in* I. Rahwan, P. Moraitis and C. Reed (eds), *Argumentation in Multi-Agent Systems*, Lecture Notes in Artificial Intelligence 3366, Springer, Berlin, Germany, pp. 162–175.

McBurney, P. and Parsons, S. 2005b. Locutions for argumentation in agent interaction protocols, *in* R. M. van Eijk, M.-P. Huget and F. Dignum (eds), *Developments in Agent Communication*, Lecture Notes in Artificial Intelligence 3396, Springer, Berlin, Germany, pp. 209–225.

McBurney, P. and Parsons, S. 2007. Retraction and revocation in agent deliberation dialogs, *Argumentation* **21**(3): 269–289.

McBurney, P. and Parsons, S. 2009. Dialogue games for agent argumentation, *in* I. Rahwan and G. Simari (eds), *Argumentation in Artificial Intelligence*, Springer, Berlin, Germany, chapter 13, pp. 261–280.

Oliva, E., McBurney, P. and Omicini, A. 2007. Co-argumentation artifact for agent societies, *in* S. Parsons, I. Rahwan and C. Reed (eds), *4th International Workshop "Argumentation in Multi-Agent Systems" (ArgMAS 2007)*, AAMAS 2007, Honolulu, Hawai'i, USA, pp. 115–130.

Oliva, E., McBurney, P. and Omicini, A. 2008. Co-argumentation artifact for agent societies, *in* S. Parsons, I. Rahwan and C. Reed (eds), *Argumentation in Multi-Agent Systems*, Vol. 4946 of *LNAI*, Springer, chapter 3, pp. 31–46. 4th International Workshop (ArgMAS 2007), Honolulu, HI, USA, 15 May 2007. Revised Selected and Invited Papers.

Oliva, E., Viroli, M., Omicini, A. and McBurney, P. 2008. Argumentation and artifact for dialogue support, *in* I. Rahwan and P. Moraitis (eds), *5th International Workshop "Argumentation in Multi-Agent Systems" (ArgMAS 2008)*, AAMAS 2008, Estoril, Portugal, pp. 24–39.

Oliva, E., Viroli, M., Omicini, A. and McBurney, P. 2009. Argumentation and artifact for dialogue support, *in* I. Rahwan and P. Moraitis (eds), *Argumentation in Multi-Agent Systems*, Vol. 5384 of *Lecture Notes in Computer Science*, Springer. Argumentation in Multi-Agent Systems, Fifth International Workshop, ArgMAS 2008, Estoril, Portugal, May 12, 2008. Revised Selected and Invited Papers.

Omicini, A. and Denti, E. 2001. From tuple spaces to tuple centres, *Science of Computer Programming* **41**(3): 277–294.

Omicini, A., Ricci, A. and Viroli, M. 2006a. *Agens Faber*: Toward a theory of artefacts for MAS, *Electronic Notes in Theoretical Computer Sciences* **150**(3): 21–36. 1st International Workshop "Coordination and Organization" (CoOrg 2005), COORDINATION 2005, Namur, Belgium, 22 April 2005. Proceedings.

Omicini, A., Ricci, A. and Viroli, M. 2006b. Coordination artifacts as first-class abstractions for MAS engineering: State of the research, *in* A. F. Garcia, R. Choren, C. Lucena, P. Giorgini, T. Holvoet and A. Romanovsky (eds), *Software Engineering for Multi-Agent Systems IV: Research Issues and Practical Applications*, Vol. 3914 of *LNAI*, Springer, pp. 71–90. Invited Paper.
URL: *http://www.springerlink.com/link.asp?id=t710627571v4256h*

Omicini, A., Ricci, A. and Viroli, M. 2008. Artifacts in the A&A meta-model for multi-agent systems, *Autonomous Agents and Multi-Agent Systems* **17**(3): 432–456. Special Issue on Foundations, Advanced Topics and Industrial Perspectives of Multi-Agent Systems.

Omicini, A., Ricci, A., Viroli, M., Castelfranchi, C. and Tummolini, L. 2004. Coordination artifacts: Environment-based coordination for intelligent agents, *in* N. R. Jennings, C. Sierra, L. Sonenberg and M. Tambe (eds), *3rd international Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, Vol. 1, ACM, New York, USA, pp. 286–293.
URL: *http://portal.acm.org/citation.cfm?id=1018409.1018752*

Oren, N. and Norman, T. 2010. Arguing using opponent models, *in* P. McBurney, I. Rahwan, S. Parsons and N. Maudet (eds), *Proceedings of the Sixth International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2009). Budapest, Hungary*, Lecture Notes in Artificial Intelligence 6057, Springer, Berlin, Germany.

Papadopoulos, G. A. and Arbab, F. 1998. Coordination models and languages, *in* M. V. Zelkowitz (ed.), *The Engineering of Large Systems*, Vol. 46 of *Advances in Computers*, Academic Press, pp. 329–400.

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

Prakken, H. and Vreeswijk, G. 2002. Logical systems for defeasible argumentation, *in* D. M. Gabbay and F. Guenther (eds), *Handbook of Philosophical Logic, Volume 4*, Kluwer, Dordrecht, pp. 219–318.

Rahwan, I., Larson, K. and Tohme, F. 2009. A characterisation of strategy-proofness for grounded argumentation semantics, *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009)*, Pasadena, CA, USA.

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

Rahwan, I. and Simari, G. (eds) 2009. *Argumentation in Artificial Intelligence*, Springer, Berlin, Germany.

Rahwan, I., Sonenberg, E., Jennings, N. R. and McBurney, P. 2007. STRATUM: a methodology for designing automated negotiation strategies, *Applied Artificial Intelligence* **21**(6): 489–527.

Reed, C. and Norman, T. J. (eds) 2004. *Argumentation Machines: New Frontiers in Argument and Computation*, Argumentation Library 9, Kluwer Academic, Dordrecht, The Netherlands.

Rehg, W., McBurney, P. and Parsons, S. 2005. Computer decision-support systems for public argumentation: assessing deliberative legitimacy, *AI and Society* **19**(3): 203–228.

Ricci, A., Viroli, M. and Omicini, A. 2008. The A&A programming model and technology for developing agent environments in MAS, *in* M. Dastani, A. El Fallah Seghrouchni, A. Ricci and M. Winikoff (eds), *Programming Multi-Agent Systems*, Vol. 4908 of *LNCS*, Springer, pp. 89–106. 5th InternationalWorkshop (ProMAS 2007), Honolulu, HI, USA, 15 May 2007. Revised and Invited Papers.

Rovatsos, M., Rahwan, I., Fischer, F. and Weiss, G. 2006. Adaptive strategies for practical argument-based negotiation, *in* S. Parsons, N. Maudet, P. Moraitis and I. Rahwan (eds), *Argumentation in Multi-Agent Systems: Second International Workshop, ArgMAS 2005 Utrecht, The Netherlands, July 26, 2005*, Lecture Notes in Computer Science 4049, Springer, Berlin, Germany, pp. 122–137.

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

Tolchinsky, P., Atkinson, K., McBurney, P., Modgil, S. and Cortes, U. 2007. Agents deliberating over action proposals using the ProCLAIM model, *in* L. Z. Varga, H.-D. Burkhard and R. Verbrugge (eds), *Multi-Agent Systems and Applications V: Proceedings of the Fifth International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS 2007). Leipzig, Germany*, Lecture Notes in Artificial Intelligence 4696, Springer, Berlin, Germany, pp. 32–41.

Tolchinsky, P., Aulines, M., Cortes, U. and Poch, M. 2009. *Advanced Agent-Based Environmental Management Systems*, Whitestein Series in Software Agent Technologies and Autonomic Computing, Birkhuser, Basel, Switzerland, chapter Deliberation about the Safety of Industrial Wastewater Discharges into Wastewater Treatment Plants, pp. 37–60.

Tolchinsky, P., Modgil, S., Cortes, U. and Sanchez-Marre, M. 2006. *Computational Models of Argument: Proceedings of COMMA 2006*, IOS Press, Amsterdam, The Netherlands, chapter CBR and Argument Schemes for Collaborative Decision Making, pp. 71–82.

Walton, D. N. and Krabbe, E. C. W. 1995. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*, State University of New York Press, Albany, NY, USA.