The Knowledge Engineering Review, Vol. 21:4, 293–316. © 2006, Cambridge University Press
doi:10.1017/S0269888906001044 Printed in the United Kingdom

Towards an argument interchange format

CARLOS CHESÑEVAR[1,6], JARRED MCGINNIS[2], SANJAY MODGIL[3], IYAD RAHWAN[2,4], CHRIS REED[5], GUILLERMO SIMARI[6], MATTHEW SOUTH[3], GERARD VREESWIJK[7] and STEVEN WILLMOTT[8]

[1]*Universitat de Lleida, Catalunya, Spain;*
e-mail: cic@eps.udl.es
[2]*University of Edinburgh, UK*
[3]*Cancer Research UK, UK*
[4]*British University in Dubai, UAE*
[5]*University of Dundee, UK*
[6]*Universidad Nacional del Sur, Argentina*
[7]*Universiteit Utrecht, The Netherlands*
[8]*Universitat Politècnica de Catalunya, Catalunya, Spain*

Abstract

The theory of argumentation is a rich, interdisciplinary area of research straddling the fields of artificial intelligence, philosophy, communication studies, linguistics and psychology. In the last few years, significant progress has been made in understanding the theoretical properties of different argumentation logics. However, one major barrier to the development and practical deployment of argumentation systems is the lack of a shared, agreed notation or 'interchange format' for argumentation and arguments. In this paper, we describe a draft specification for an argument interchange format (AIF) intended for representation and exchange of data between various argumentation tools and agent-based applications. It represents a consensus 'abstract model' established by researchers across fields of argumentation, artificial intelligence and multi-agent systems[1]. In its current form, this specification is intended as a starting point for further discussion and elaboration by the community, rather than an attempt at a definitive, all-encompassing model. However, to demonstrate proof of concept, a use case scenario is briefly described. Moreover, three concrete realizations or 'reifications' of the abstract model are illustrated.

1 Introduction and background

The theory of argumentation is a rich, interdisciplinary area of research straddling the fields of philosophy, communication studies, linguistics and psychology. Its techniques and results have found a wide range of applications in both theoretical and practical branches of artificial intelligence and computer science, as outlined in various recent reviews (Carbogim:*et al.*, 2000; Chesñevar *et al.*, 2000; Prakken & Vreeswijk, 2002, Rahwan *et al.*, 2003; Reed & Norman, 2004). These applications involve a wide range of areas such as specifying semantics for logic programs (Dung, 1995), natural language processing (Elhadad, 1995), recommender systems technology (Chesñevar *et al.*, 2006), legal reasoning (Bench-Capon, 1997), decision-support for multi-party human decision-making (Gordon & Karacapilidis, 1997), conflict resolution (Sycara, 1992), as well as recent applications in multi-agent systems (Rahwan, 2005; Rahwan *et al.*, 2005; Parsons *et al.*,

[1] This article started life as a skeleton for contributions from participants at an AgentLink Technical Forum Group meeting in Budapest in September 2005, receiving additional input from third parties. The results were subsequently improved and added to by online discussion to form a more substantial consensus.

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press

2006) in both the autonomous reasoning of individual agents (Kakas & Moraitis, 2003)—and particularly defeasible reasoning (García, 2004)—and the structure of interactions between them (McBurney & Parsons, 2002).

While significant progress has been made in understanding the theoretical properties of different argumentation logics (Prakken & Vreeswijk, 2002) and in specifying argumentation dialogues (McBurney & Parsons, 2003), there remain major barriers to the development and practical deployment of argumentation systems. One of these barriers is the lack of a shared, agreed notation or 'interchange format' for argumentation and arguments. In the last few years a number of different argument mark-up languages (AMLs) have been proposed in the context of tools developed for argument visualization and construction (see Kirschner *et al.* (2003) for a review). Thus, for example, the Assurance and Safety Case Environment (ASCE)[2] is a graphical and narrative authoring tool for developing and managing assurance cases, safety cases and other complex project documentation. ASCE relies on an ontology for arguments about safety based on claims, arguments and evidence (Emmet & Cleland, 2002). Another mark-up language was developed for Compendium[3], a semantic hypertext concept mapping tool. The Compendium argument ontology enables constructing issue based information system (IBIS) networks, in which nodes represent issues, positions and arguments (Conklin & Begeman, 1988).

The analysis and study of human argument has also prompted the development of specialized AMLs and tools. Two particularly relevant developments in this direction are *ClaiMaker* (Shum *et al.*, 2006) and AML (Reed & Rowe, 2004). *ClaiMaker* and related technologies (Shum *et al.*, 2006) provide a set of tools for individuals or distributed communities to publish and contest ideas and arguments, as is required in contested domains such as research literatures, intelligence analysis or public debate. This system is based on the *ScholOnto* ontology (Shum *et al.*, 2000), which can express a number of basic reasoning schemes (causality, support) and relationships between concepts found in scholarly discourse (e.g. similarity of ideas, taxonomies of concepts, etc.). The AML used by the Araucaria system[4] is an XML-based language (Reed & Rowe, 2004) designed for the markup of analysed human argument. The syntax of the AML is specified in a document type definition (DTD) which imposes structural constraints on the form of valid AML documents. The AML was primarily produced for use in the Araucaria tool, although it has more recently been adopted elsewhere.

These various attempts at providing AMLs share two major limitations. First, each particular language is designed for use with a specific tool (usually for the purpose of facilitating argument visualization) rather than for facilitating inter-operability of arguments among a variety of tools. As a consequence, the semantics of arguments specified using these languages is tightly coupled with particular schemes to be interpreted in a specific tool and according to a specific underlying theory. Thus, for example, arguments in the Compendium concept mapping tool are to be interpreted in relation to a rigorous theory of issue-based information systems. Clearly, in order to enable true inter-operability of arguments and argument structures we need an argument description language that can be extended beyond a particular argumentation theory, enabling us to accommodate a variety of argumentation theories and schemes. Another limitation of the above AMLs is that they are primarily aimed at enabling users to structure arguments through diagrammatic linkage of natural language sentences (Kirschner *et al.*, 2003). Hence, these mark-up languages are not designed to process formal logical statements such as those used within multi-agent systems. For example, AMLs impose structural limitations on legal arguments, but provide no semantic model. Such a semantic model is a natural requirement in order to enable the automatic processing of argument structures by software agents.

In order to address these limitations, a group of researchers interested in 'argument and computation' gathered for a workshop[5] whose aim was to sketch an argumentation interchange

[2] `http://www.adelard.co.uk/software/asce/`.
[3] `http://www.compendiuminstitute.org/tools/compendium.htm`.
[4] `http://araucaria.computing.dundee.ac.uk/`.
[5] AgentLink Technical Forum Group meeting, Budapest, Hungary, September 2005.

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press

format (AIF) which consolidates—where possible—the work that has already been done in AMLs and multi-agent system frameworks. The main aims of the AIF were as follows:

1. to facilitate the development of (closed or open) multi-agent systems capable of argumentation-based reasoning and interaction using a shared formalism;
2. to facilitate data interchange among tools for argument manipulation and argument visualization.

In this paper, we describe and analyse the main components of a draft specification for AIF. It must be remarked that AIF as it stands represents a consensus 'abstract model' established by researchers across fields of argumentation, artificial intelligence and multi-agent systems. In its current form, this specification is intended as a starting point for further discussion and elaboration by the community, rather than an attempt at a definitive, all-encompassing model. In order to demonstrate the power of the proposed approach, we describe use cases which show how AIF fits into some argument-based tools and applications. We also illustrate a number of concrete realizations or 'reifications' of the proposed abstract model.

The rest of this paper is structured as follows. In Section 2 we describe our overall approach to the development of an AIF. The core abstract AIF model describing basic concepts and their relationships is described in Section 3. Then, in Section 4 we describe use cases illustrating how AIF fits into some argument-based tools and applications in the context of the ASPIC project[6]. Section 5 then describes three particular reifications/syntaxes instantiating the model described in Section 3. Finally, in Section 6 we present the main conclusions obtained and discuss some open issues.

2 Overall approach

An AIF, like any other data representation, requires well-defined syntax and semantics. The syntax is required as a concrete representation of statements relating to arguments, and the semantics conveys the meaning of statements made using the syntax. However, beyond this basic requirement, there are a wide range of approaches which could be taken for defining both syntax and semantics. In particular, semantics may be explicit (using some previous formal notation with its own syntax and semantics) or implicit (hard coded into a piece of software which subsequently behaves in a given way for each combination of inputs), machine readable or targeted at a human audience (written notes for human consumption), formal or informal, etc. Further questions arise as to whether there should be one single AIF defined, whether variations should be allowed for, how extensions should be dealt with, etc. Given this range of possibilities the approach taken in this document adheres to the following overall principles.

1. Machine readable syntax. AIF representations are specifically targeted at machine read/write operations rather than human-level documentation. While using formats which are human readable is desirable (e.g. for debugging purposes), the primary aim of the format is data interchange between software systems.
2. Explicit and (where possible) machine processable semantics. First, the semantics of AIF statements are to be stated explicitly in specification documents, so that such statements can be implemented by multiple tool/system providers. Second, where possible, the nature of the semantic definition should enable the implementation of processing tools such as reasoners (e.g. using some existing logical framework).
3. Unified abstract model, multiple reifications. The AIF should be defined in terms of (a) an abstract model characterizing the concepts which could be expressed in an AIF and their relationship to one other, and (b) a set of concrete reifications/concrete syntaxes which instantiate these concepts in a particular syntactic formalism (such as XML, Lisp-like S-expressions, etc.). That way, inter-operability may still be facilitated by similarities at the

[6] `http://www.argumentation.org`.

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press

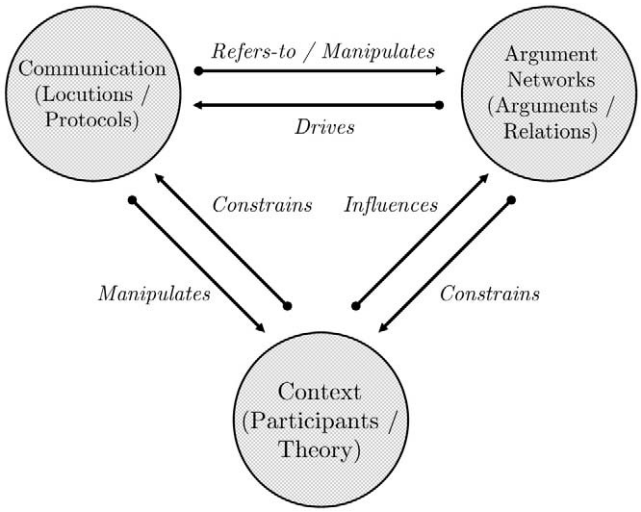


Figure 1 Overview diagram of main groups of concepts defined by the AIF Core Ontology

abstract level even when dealing with different computational environments with particular syntactic requirements.

4. Core concepts, multiple extensions. Recognizing that different applications may require statements about a wide array of different argumentation related concepts, the AIF will be structured as a set of core concepts (i.e. those likely to be common to many applications) and extensions (those which are specialist to particular domains or types of applications). It is anticipated that (a) the core will evolve over time as consensus changes on what is central and applications generate experience, and that (b) extensions could be generated by any user of the AIF and, if they turn out to be particularly useful, would be shared amongst large groups of users (potentially also being merged into the core).

3 Abstract model/core ontology

The foundation for the AIF model is given by a set of definitions for high-level concepts related to argumentation, which may need to be represented in the proposed format. These concepts are gathered into three main groups, as follows:

1. Arguments and argument networks. The core ontology for argument entities and relations between argument entities with the purpose of reification in an AIF (see Section 3.2).
2. Communication. The core ontology for items which relate to the interchange of arguments between two or more participants in an environment, including locutions and protocols (see Section 3.4).
3. Context. The core ontology for items associated with environments in which argumentation may take place. These include participants in argument exchanges (agents), theories contained in the environment that are used for argumentation, and other aspects which may affect the meaning of arguments/communication of arguments (see Section 3.5).

In the following subsections an overview of the above concepts is given. Definitions are drawn from existing theories when possible, but may diverge where alignment among theories is needed. The relationships between these groups of concepts are shown in Figure 1.

3.1 The notion of argument

Before proceeding with these definitions, it is worth noting that we will not take a position on the precise definition of the notion of 'argument' itself, even though later sections do provide structures

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press

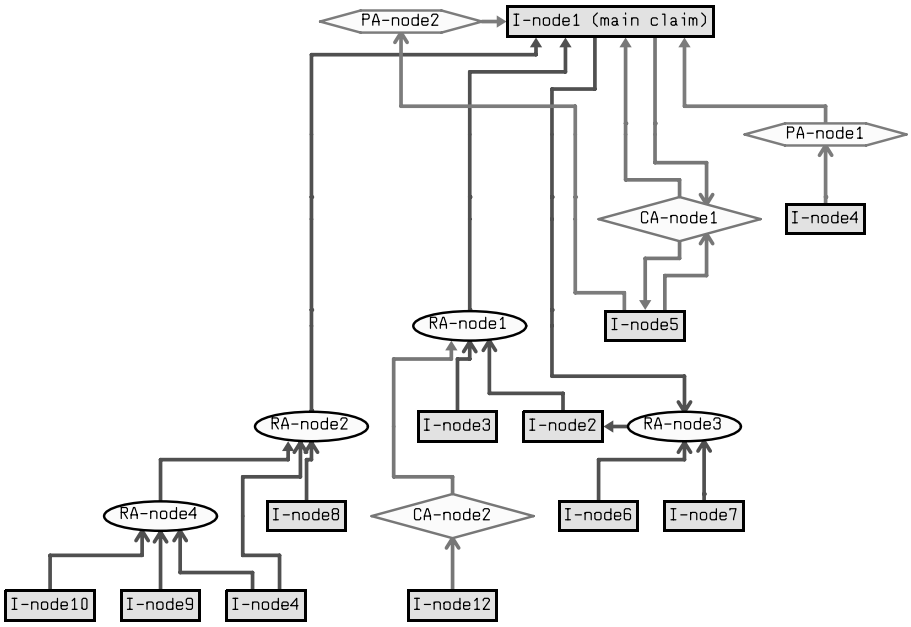


Figure 2 Sample AN

for describing argument. The reason for this is that initially we found it too difficult to select a single definition acceptable to all. We contend that progress on such a definition might be better made once some consensus is reached on the necessary lower level concepts. A useful starting point for understanding philosophical notions of argument can, however, be found in David Hitchcock's input to the original AIF meeting[7].

3.2 Arguments/argument networks

The following section defines the top-level concepts to be considered for an ontology of arguments and relationships between arguments.

3.2.1 Concepts and relations

The starting point of this section is the assumption that argument entities can be represented as nodes in a directed graph (di-graph). This di-graph is informally called an argument network (AN). An example of an AN is displayed in Figure 2. The rationale for not restricting ourselves to directed acyclic graphs (DAGs) or even trees is that argumentation formalisms vary to a great extent. A number of formalisms allow for cycles where others forbid them explicitly. One of our basic assumptions is that the core ontology should cater for these differences, and should be able to capture extreme cases.

3.2.2 Nodes

There are two types of node, namely information nodes (I-nodes) and scheme application nodes or scheme nodes (S-nodes) for short (see Figure 3).

Whereas I-nodes relate to content and represent claims that depend on the domain of discourse, S-nodes are applications of schemes. Such schemes may be considered as domain-independent patterns of reasoning (which resemble rules of inference in deductive logics but broadened to include non-deductive logics that are not restricted to classical logical inference). The present ontology deals with three different types of schemes, namely inference schemes, preference schemes and conflict schemes. Potentially, other scheme types could exist, such as evaluation schemes and

[7] http://www.x-opennet.org/aif/Inputs/aif2005_david_hitchcock_1.pdf.

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press

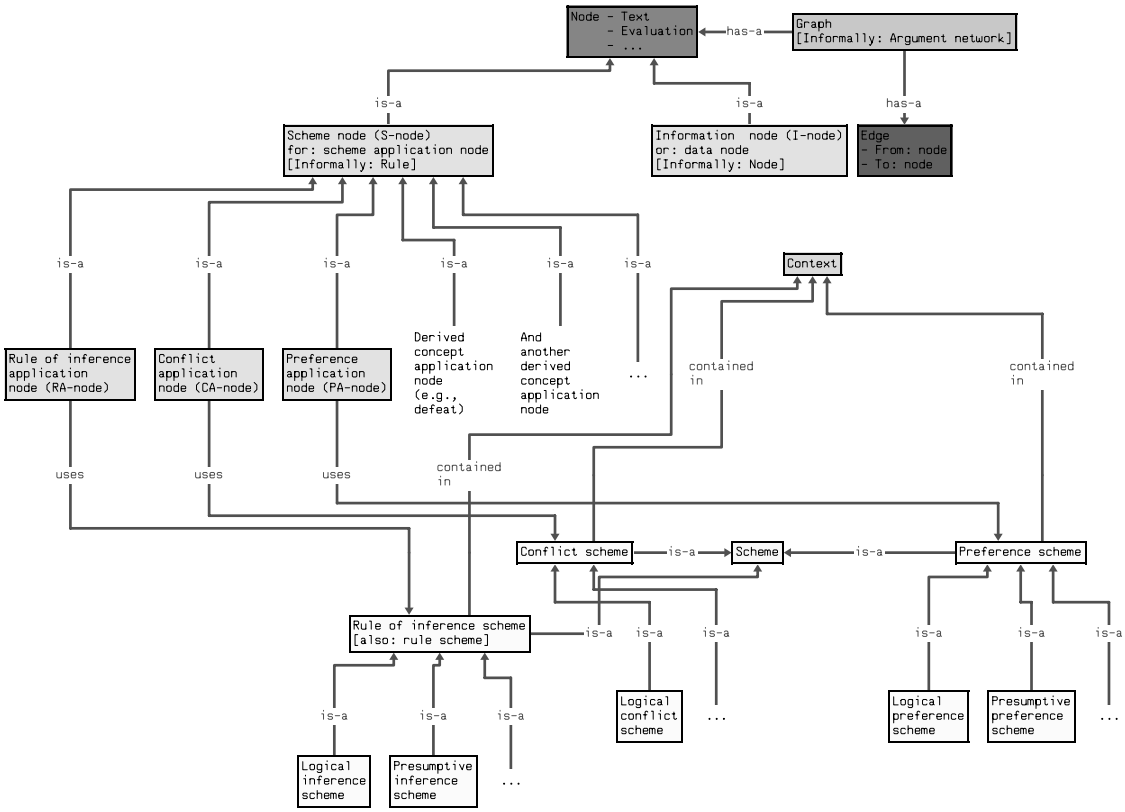


Figure 3 Concepts and relations for an ontology of arguments

scenario schemes, which will not be addressed further here. Notice that presumptive argumentation schemes, such as those presented by (Walton, 1996), constitute a subset of the set of possible inference schemes.

If a scheme application node is an application of an inference scheme, it is called a rule of inference application node (RA-node). If a scheme application node is an application of a preference scheme it is called a preference application node (PA-node). Similarly, if an S-node is an application of a conflict scheme, it is called a conflict application node (CA-node). Informally, RA-nodes can be seen as applications of (possibly non-deductive) rules of inference, whereas CA-nodes can be seen as applications of criteria (declarative specifications) defining conflict, which may be logical or non-logical. PA-nodes, however, are applications of (possibly abstract) criteria of preference among evaluated nodes.

3.2.3 Node attributes

Nodes may possess different attributes such as `title, creator, type` (e.g. decision, action, goal, belief), `creation date, evaluation` (or `strength`, or `conditional evaluation table`), `acceptability`, and `polarity` (e.g. values such as 'pro' or 'con'). These attributes may vary and are not part of the core ontology. The term 'conditional evaluation table' is inspired by its Bayesian analogue called 'conditional probability table' (CPT), and may be used to capture information useful in evaluating individual arguments or groups of arguments. Most attributes are proper, that is, essential to the node itself, while others are derived. It is imaginable that a derived attribute such as `acceptability` may be obtained from node-specific attributes through calculation. In this case, the `acceptability` of an argument may be obtained from `evaluation` through mechanical inference[8].

[8] There are voices that advocate dropping derived node attributes altogether, for different algorithms may assign different values for such attributes to arguments within one and the same AN.

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press

Table 1 Semantics of support for node-to-node relationships in an AN

	To I-node	To RA-node	To PA-node	To CA-node
From I-node		I-node data used in applying an inference	I-node data used in applying a preference	I-node data in conflict with information in node supported by CA-node
From RA-node	Inferring a conclusion in the form of a claim	Inferring a conclusion in the form of an inference application	Inferring a conclusion in the form of a preference application	Inferring a conclusion in the form of a conflict definition application
From PA-node	Applying a preference over data in I-node	Applying a preference over inference application in RA-node	Meta-preferences: applying a preference over preference application in supported PA-node	Preference application in supporting PA-node in conflict with preference application in PA-node supported by CA-node
From CA-node	Applying conflict definition to data in I-node	Applying conflict definition to inference application in RA-node	Applying conflict definition to preference application in PA-node	Showing a conflict holds between a conflict definition and some other piece of information

3.2.4 Edges

In the context of a graph representing argument-based concepts and relations, a node A is said to support node B if and only if there is an edge running from A to B. Edges do not need to be explicitly marked, labelled, or otherwise supplied with semantic pointers. If desired, edge types can be inferred from the nodes they connect. Basically, there are two types of edge, namely scheme edges and data edges. Scheme edges emanate from S-nodes and are meant to support conclusions that follow from the S-node. These conclusions may either be I-nodes or S-nodes. Data edges emanating from I-nodes, however, necessarily end in S-nodes, and are meant to supply data, or information to scheme applications. In this way, we can speak of I-to-S edges ('information' or 'data' supplying edges), S-to-I edges ('conclusion' edges) and S-to-S edges ('warrant' edges).

Table 1 summarizes the relations associated with the semantics of support. Notice that I-to-I edges are forbidden, because I-nodes cannot be connected without an explanation for why that connection is being made. There is always a scheme, justification, inference or rationale behind a relation between two or more I-nodes that is captured in some form of S-node. Moreover, only I-nodes can have zero incoming edges, as all S-nodes relate two or more components (for RA-nodes, at least one antecedent is used to support at least one conclusion; for PA-nodes, at least one alternative is preferred to at least one other; and for CA-nodes, at least one claim is in conflict with at least one other).

Some further explanation is required with regard to S-to-S edges. These allow us to represent what might more properly be considered as modes of 'meta-reasoning'. For example, RA-to-RA and RA-to-PA edges might indicate some type of meta-justification for application of an inference rule or particular criterion for defining preferences. Some instances of Toulmin backings (Toulmin, 1958), for example, could most accurately be captured through the use of RA-to-RA links. An RA-to-CA node could encode some rationale for why two I-nodes are in conflict; for example, that each I-node specifies two alternative actions for realizing a goal (in which case, arguments supporting each action are considered to be in conflict). Of course, once we consider these forms of meta-reasoning, then this paves the way for 'meta-argumentation' in that two preference

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press

applications might be in conflict (PA-to-CA and CA-to-PA), requiring the definition of a preference between preference applications (PA-to-PA; Modgil, 2006).

To distinguish scheme edges from data edges in diagrams, edges that emanate from S-nodes may be supplied with a closed arrowhead at the end, while edges that emanate from I-nodes may be supplied with an open arrowhead at the end. Edges may be further classified into different categories, such as support edges (that are associated or 'coloured' by the scheme of the S-node they are connected to; for S-to-S edges, the nodes that they emanate from), inference edges (those edges that are connected to a RA-node, shown in black in Figure 2) and attack edges (edges that are connected to a PA-node, shown in red in Figure 2)[9]. Marking edges and applying arrowheads to edges is not part of the ontology but only meant to help human beings in its interpretation.

3.2.5 Derived concepts

Concepts from an extension ontology, in particular concepts such as rebut, undercut, attack, defeat and defend, can in principle be derived from the concepts in the diagram that is displayed in Figure 3. Thus, an argument qualified with derived concepts can in principle be described in terms of basic concepts in a mechanical manner. Nevertheless, such derived concepts may still have an important place that should be respected by their inclusion in a suitable extension ontology (see further discussions in Section 6).

3.3 Examples

This section presents three examples: an abstract example that shows most of the features of the ontology, a translation of the Toulmin scheme and a simple concrete example.

3.3.1 Abstract example of an AN

An abstract example of an AN is displayed in Figure 2. This network contains 11 I-nodes, namely `I-node1`, ..., `I-node11` and six rule application nodes, namely `RA-node1`, `RA-node2`, `RA-node3`, `RA-node4`, `PA-node1` and `PA-node2`. This abstract example is meant to demonstrate the flexibility of the core ontology, stretching the limits of the model. Obviously, many existing argument formalisms would not support the constructions shown in this example. Some observations that can be drawn from the diagram are the following.

1. The main claim is supported by two inference applications, two preference applications and one conflict application.
2. `I-node4` shows that our model allows for multiple node references. Thus, nodes may be referred to more than once. Consequently, ANs are not restricted to tree-like structures as those used for modelling argument games or dialectical proof procedures (Amgoud & Cayrol, 2002}.
3. The two inference applications

   `I-node2, I-node3-(RA-node1)→ I-node1` (main claim)

   `I-node1, I-node6, I-node7-(RA-node3)→ I-node2`

   show that, in theory, cycles may occur.
4. `CA-node1` links and supports both `I-node1` and `I-node5` reflecting the symmetrical nature of the conflict (commonly known as a rebut attack in the literature). Also, `PA-node2` emanating from `I-node5` and supporting `I-node1` indicates preference for the former over the latter. One might therefore obtain the derived relationship that `I-node5` defeats `I-node1`, given that it conflicts with (attacks) and is preferred to `I-node1`.
5. `RA-node1` is asymmetrically attacked by `I-node12` via `CA-node2`. An attack on an inference application is often referred to as an undercut (Pollock, 1987). However, a CA-node may also

[9] Note that in the colour printed version of the document different colours are visible for edges for clarity; however, they are not essential to interpretation.

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press

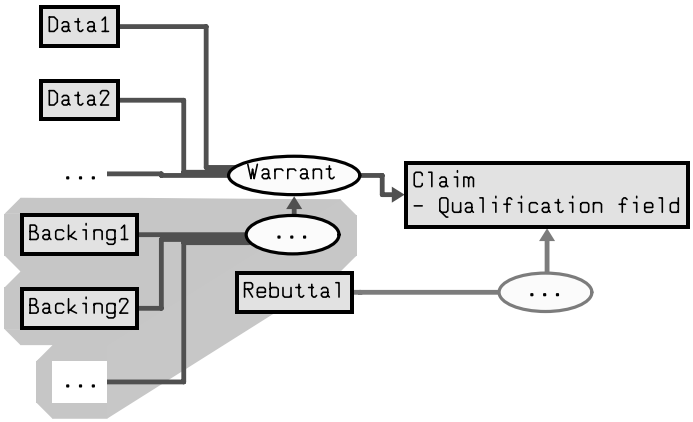


Figure 4 Toulmin scheme conceptualized as an AN

asymmetrically support (i.e. attack) an I-node. Such attacks are also referred to as undercuts in the literature; for example, in (Prakken & Sartor, 1997) an argument *A* undercuts another argument *A″* if *A* proves (claims) what was assumed unprovable by A′.

3.3.2 *Argumentation à la Toulmin: an example*

The scheme presented by Toulmin (Toulmin, 1958} has been very influential in the computational modelling of argument. In its simplified form, the Toulmin scheme consists of six essential elements: data (*D*), warrant (*W*), backing (*B*), qualifier (*Q*), rebuttal (*R*) and claim (*C*). These elements are usually depicted as follows:

$$D \longrightarrow Q,\ C$$
$$\text{since } W \quad \text{unless } R$$
$$B$$

A (somewhat liberal) translation of this scheme into our AIF format is displayed in Figure 4. The shadow-encircled nodes together relate to the original backing *B*. Notice that in Figure 4, *R* (rebuttal) attacks *C* (claim) rather than *W* (warrant). It is not clear from Toulmin (1958) whether *R* should attack *C* or *W*. In our approach to AIF we have chosen the latter. Nevertheless, *R* can reasonably be taken to attack *C*, to support not-*C*, to attack *W* or to attack an implicit warrant. In this particular case our AIF approach does not advocate a specific mechanism for such translation, but merely that any of those translations should be representable in the present ontology.

3.3.3 *A concrete and simple example*

In Figure 5 we show a concrete and simple example of an AN for handling the well-known AI example of modelling the flying abilities of birds and penguins, and reasoning about whether a particular penguin `opus` can fly. In this case there are two arguments, one for `fly(opus)` and one for `~fly(opus)`, where '~' denotes negation.

The argument for `~flies(opus)` is composed of one scheme-application, namely *Modus Ponens* (MP). A simplistic version of MP reads as follows: *if there are two information nodes* `A(x)` *and* `A(x)->B(x)` *then conclusively infer* `B(x)`. The argument for `flies(opus)` is composed of one scheme-application, namely *defeasible Modus Ponens* (dMP). A simplistic version of dMP reads as follows: *if there are two information nodes* `A(x)` *and* `A(x)-(qualifier)->B(x)` *then defeasibly infer* `B(x)`. The conflicting nodes `flies(opus)` and `flies(opus)` are related by an intermediate CA-node linking the conflicting nodes in both directions. The argument for `flies(opus)` is

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press

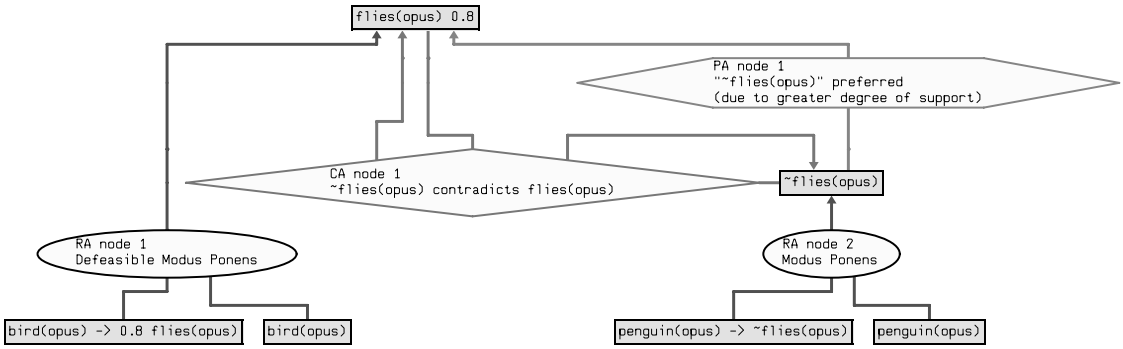


Figure 5 Modelling arguments 'pro' and 'con' flying abilities in birds: a concrete example of an AN

conclusive and therefore preferred to (and so defeats) the argument for `flies(opus)`. Hence the intermediate PA-node linking `flies(opus)` to `flies(opus)`.

3.4 Communication: locution/protocols

The second group of concepts correspond to those which concern communication in the context of argumentation, such as, for example, concepts which capture:

1. the utterance of a statement by an agent containing an argument or AN;
2. a sequence of legal statements making reference to arguments/ANs which could be made by a set of agents in order to make a decision or reach some other goal.

In turn, as with arguments/ANs, communication also takes place in a context, elements of which may affect the interpretation of statements (such as references to the participants in a dialogue, the ontologies applying, the semantic models adopted, etc.). In this setting, two main elements can be identified: locutions, which are individual words, phrases or expressions uttered by an agent, and interaction protocols, which are defined by sequences of locutions involving one or more (usually at least two) agents and normally designed to achieve a specific goal (such as reaching an agreement or giving information). Although locutions provide the basic building blocks of protocols, it is important to note that there are different 'schools of thought' on how the semantics for locutions and protocols should be defined in terms of one another. One approach, such as FIPA (FIPA, 2001), holds that semantics are attributed to individual locutions, and the semantics of a protocol are a compound of the semantics of individual locutions. Another approach holds that the semantics of locutions vary depending on their context (e.g. the commitments made thus far) and hence their place in a particular protocol (Maudet & Chaib-draa, 2002; Reed, 2006).

3.4.1 Locutions

A rich literature exists on locutions of various types and their semantics. In terms of general agent communication, languages such as FIPA-ACL and KQML define sets of general locutions such as *inform, request, query, tell* and so on, each with an associated formal logical semantics. However, while these languages may provide useful resources, it is also clear from more specific argumentation literature that the types of locutions which occur frequently are more specific than (or different to) those found in FIPA-ACL/KQML. Typical examples include *assert, accept, challenge, question, concede* and *prefer*.

While different authors use different labels for different locutions, there seem to often be similarities in semantics. Work such as that by McBurney (2002), McBurney & Parsons (2002)}, Maudet & Chaib-draa (2002) and McBurney, 2005) provides a starting point for potentially determining a limited number of locutions which could form the core of an AIF, others potentially being added as extensions. In this setting, an AIF core ontology should include the generic notion

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press

of locution and distinguish a set of individual locutions, formed by a set of subclasses of the class of locutions listed at the beginning of this section. Locutions will have a number of associated properties. Based on the FIPA-ACL message structure specification (FIPA, 2001), such properties might include[10] the following.

1. *Sender*: the agent uttering a locution (note that a distinction could also be made between the sender who makes an utterance and the originator(s)—an agent or group of agents responsible for generating the utterance).
2. *Receiver* or *receivers*: agents 'hearing' an utterance (distinctions could be made between intended recipients, those intentionally made aware of the message but not the intended recipients and those who unintentionally become aware of an utterance).
3. *Ontologies*: the ontologies which define elements of the content.
4. *Language*: the content language used in the content part of the message (which should itself have a formal semantics).
5. *Protocol*: the protocol a locution is part of.
6. *Content*: the object of the locution.
7. *Message management elements*: items such as a message-identifier, a conversation-identifier, in-reply-to field, etc.

3.4.2 Interaction protocols

It is possible to construct comprehensive standards of language usage for computational systems that are widely used and relatively precise. This is the case for programming language standards (such as PROLOG, dialects of ADA, etc.). By contrast, in areas where standardization of more abstract concepts is required, consensus appears to be much harder to achieve, because abstract concepts are difficult to pin down uniquely in a simple way. In this circumstance it is often expedient to define precisely a core standard, containing only those elements essential to getting the job done, and then allow extensions to this core in a controlled (but perhaps less precise) way. An example of this form of standardization is the process interchange format (PIF), which is a standard for describing processes. The PIF core contains a small number of very generic concepts at the heart of that standard allows those with specific process description needs to meet their own requirements by building on that core.

The definition of an interaction protocol language as part of an AIF provides a number of advantages. If the protocol language can be used for computation then it can be effectively considered to be a programming standard, and history suggests that such standards tend to be durable because they connect to practice (or fail to connect and then die cleanly). If it is also declarative—and hence independent of current fashion in low-level implementation languages or basic communications protocols—then it can support formal analysis and verification more readily. In addition, the use of a high-level language arguably facilitates human readability. For software engineers there is a natural notion of pattern in the design of protocols and this is one approach to extension from a core protocol syntax to a (more interesting) set of extensions via patterns.

The design and analysis of protocols is an area where traditionally computer science has helped to supply standards. For example, Figure 6 defines the syntax of the lightweight coordination calculus (LCC) that uses a combination of traditional specification drawn from CCS and logic programming (for details on LCC, see Robertson (2004)). An interaction model in LCC is a set of clauses, each of which defines how a role in the interaction must be performed. Roles are described by the type of role and an identifier for the individual agent undertaking that role. The definition of performance of a role is constructed using combinations of the sequence operator ('then') or choice operator ('or') to connect messages and changes of role. Messages are either outgoing to another agent in a given role ('$\Rightarrow$') or incoming from another agent in a given role ('$\Leftarrow$'). Message input/output or change of role can be governed by a constraint defined using the normal logical operators for conjunction, disjunction and negation. Notice that there is no commitment to the system of logic through which

[10] Note that additionally one could add a slot for semantics which points to the defined formal semantics for the locution.

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press

Model	:=	[*Clause*, . . .]
Clause	:=	*Role* :: *Def*
Role	:=	a(*Type, Id*)
Def	:=	*Role* | *Message* | *Def then Def* | *Def or Def*
Message	:=	$M \Rightarrow Role \mid M \Rightarrow Role \leftarrow C \mid M \Leftarrow Role \mid C \leftarrow M \Leftarrow Role$
C	:=	*Constant* | *P*(*Term*, . . .) | $\neg C \mid C \wedge C \mid C \vee C$
Type	:=	*Term*
Id	:=	*Constant* | *Variable*
M	:=	*Term*
Term	:=	*Constant* | *Variable* | *P*(*Term*, . . .)
Constant	:=	lower case sequence or number
Variable	:=	upper case character sequence or number

Figure 6 LCC syntax

constraints are solved; on the contrary, we would expect different agents to operate different constraint solvers. Hence the standardization in LCC is on the generic language for describing interaction (only) and in this sense it is 'core'. This standardization also has the added benefit of having a style of description that is close to computation (which except for the process operators is quite close to logic programming, where there already exists a successful ISO standard).

3.5 Context: general context/participants/theory

The third group of concepts in the ontology is that of elements which form the context in which argumentation takes place. In keeping with the distinction already made between concepts for communication and those for arguments/ANs, concepts related to context may also be usefully grouped into these two areas.

3.5.1 Communication context

Here, context captures information relevant to argument-based dialogues. These include the following.

1. *Participants*: characterizing participants in argument-based dialogues may require references to agents taking part in the dialogue, possibly including the following:
   a Participant ID: an identifier for a participant.
   b Participant role: the role of the participant in relation to the dialogue (e.g. pro, con, persuader, buyer, seller, etc.). This may influence the way dialogue proceeds.
2. *Dialogue topic*: this refers to the main issue under discussion (e.g. the question under enquiry, or resource under negotiation).
3. *Dialogue type*: a reference to the type of the dialogue (e.g. persuasion, negotiation (Walton & Krabbe, 1995)). This can be simply a name, or it can be a pointer to a more elaborate dialogue typology.
4. *Background theory*: this includes statements that participants agree upon (e.g. legal rules), and which may be used to construct arguments within the dialogue.
5. *Commitment stores*: this is a data structure that allows agents to add and remove commitments during their dialogues (Hamblin, 1970).
6. *Commitment rules*: these are rules that specify how dialogue participants may modify the content of commitment stores.

3.5.2 AN context

Here, context captures information relevant to the interpretation and processing of the AN.

1. *Argumentation theory rules*. These are the rules that specify the way arguments are constructed and interpreted. In a way, they represent the underlying formal argumentation theory including the following:

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press

Inference rules. These can be thought of as the specifications of the types of inference application nodes that can be used in the AN.
Conflict rules. These can be thought of as domain-oriented declarative specifications of the notion of conflict between two pieces of information.
Preference rules. Similarly, these can be thought of as the specifications of the types of preference application nodes that can be used in the AN.

2. *Background theory*. This includes statements taken for granted (e.g. legal rules), and which may be used to interpret or process arguments.
3. *Domain ontologies*. Additionally, references to ontologies could be added to interpret ANs. For example, suppose an AN represents claims and justifications of the medical properties of a particular drug. In order to process these arguments automatically, we may benefit from a specialized medical drug ontology while interpreting these arguments.

4 Some use cases for AIF

In this section we describe different use cases which illustrate how our AIF proposal fits into the broader landscape of tools and applications. Here we briefly present use case scenarios for an XML reification recently developed for the ASPIC (Argumentation Service Platform with Integrated Components) project[11]. ASPIC aims at development of theoretical models for: (a) formal logic-based inference of argument; (b) decision-making based on arguments for and against decision options; (c) protocol-based use of arguments so as to enrich purposive dialogues between agents; (d) integration of argumentation with machine learning. ASPIC also aims to transition the aforementioned theoretical models to rigorously engineered generic software components, and provide a platform for application developers to integrate ASPIC components into standalone applications and agent technologies.

Currently, within the ASPIC project an inference component is available[12] based on a prototype developed by Gerard Vreeswijk, which is intended for deployment in agent applications and linkage to argument visualization and editing tools. After describing the features of the inference component, we then briefly describe some use cases illustrating linkage of the component to visualization and editing tools, and in so doing highlight requirements for a reification of the AIF concepts described in Section 3.2.

4.1 The ASPIC inference component

The ASPIC inference component can conceptually be broken down into four Argumentation Modules (ArMs) implementing the following:

1. construction of tree structured arguments (*ArgCon*) from a knowledge base of facts and strict and defeasible rules;
2. argument valuation (*ArgVal*), currently based on the 'weakest link' principle, but intending to offer a range of additional criteria for argument strength valuation (e.g. 'last link');
3. definitions of argument interaction (*ArgInt*); in particular *attack* and *defeat*, where the latter additionally accounts for the relative strengths of arguments;
4. evaluation of the dialectical status of arguments (*ArgStat*) on the basis of the ways in which they interact. In particular, the component implements algorithms based on argument games for determining Dung acceptability under grounded and preferred semantics (Dung, 1995).

[11] ASPIC (IST-002307) is an Integrated Project of the European Union's 6th Framework (`http://www.argumentation.org`).
[12] See `http://aspic.acl.icnet.uk/` for more details.

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press

4.2 Linking the component to argument structuring and visualization tools

A number of tools have been developed for construction and visualization of arguments including the following:

1. diagrammatic/graphical argument structuring and visualization tools in professional, research and pedagogic domains, focusing on the advantages argumentation brings to the process of capturing human reasoning and debate (e.g. Araucaria (Reed & Rowe, 2004) and ArguMed (Verheij, 1999), among others);
2. argument structuring and elicitation tools enabling users to incorporate their individual knowledge via argumentation and support users by providing access to further domain content for their arguments (e.g. Room 5 (Loui, 1997));
3. systems aiming at machine-authoring effective arguments from knowledge bases which are presented to the human user, via reasoned and persuasive explanations (e.g. PLAID (Bench-Capon & Staniford, 1995));
4. collaborative decision-making tools aiming at improving deliberation amongst multiple stakeholders; (e.g. Conklin & Begeman, 1988). Essentially, these systems enable construction of argument graphs through dialectical interaction amongst multiple users. These graphs embed answers to the hows and whys behind the decisions arrived at.

A useful, unifying perspective on the above tools might be to describe them all as employing argumentation as 'a framework to support the user's interaction with knowledge'. This involves:

1. knowledge elicitation from the user or knowledge retrieval from an external knowledge source;
2. enhancing the explanatory power of knowledge through argument;
3. guidance in construction of arguments;
4. visualization of arguments to aid understanding of the dialectical relationships between arguments.

It is in each of these areas that deployment of the inference component can enhance existing functionality and provide novel functionality. We describe some uses below, and in so doing highlight requirements for a reification of the AIF.

1. Two modes of elicitation of 'atomic' knowledge for argument construction are envisaged: first, retrieval from external knowledge repositories (arrow 1 in Figure 7) including relational and deductive databases, ontologies, Web-accessible knowledge services, and other agents; second, visualization and editing tools can be used to elicit entry of 'atomic argumentation knowledge' (the basic building blocks for argument construction) by a human user (arrow 2 in Figure 7). In both cases, the input must be mapped to some standard representation format for atomic argumentation knowledge which is subsequently communicated to the knowledge base (arrow 4 in Figure 7) from which *ArgCon* constructs arguments.
2. Linkage of the *ArgCon* argumentation module to external knowledge bases (repositories) will facilitate argument structuring of system explanations. For example, consider an expert system that outputs an explanation (reasoning trace) of a successful PROLOG query to a PROLOG knowledge base. Linkage of this knowledge base to the *ArgCon* module (arrow 3), together with an appropriate mapping of the PROLOG encoded knowledge to the common representation format, will facilitate the appropriate rendering of the explanation as an argument (arrow 5). Such rendering could be textual or also graphical (if linked to an argument visualization tool).
3. Linkage of the *ArgCon* argumentation module to visualization tools can also serve to guide users to construct valid arguments. For example, the *ArgCon* module can guide a user (arrow 5) to enter natural language sentences instantiating an informal presumptive argument scheme (arrow 6) of the type described in Walton (1996).

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press

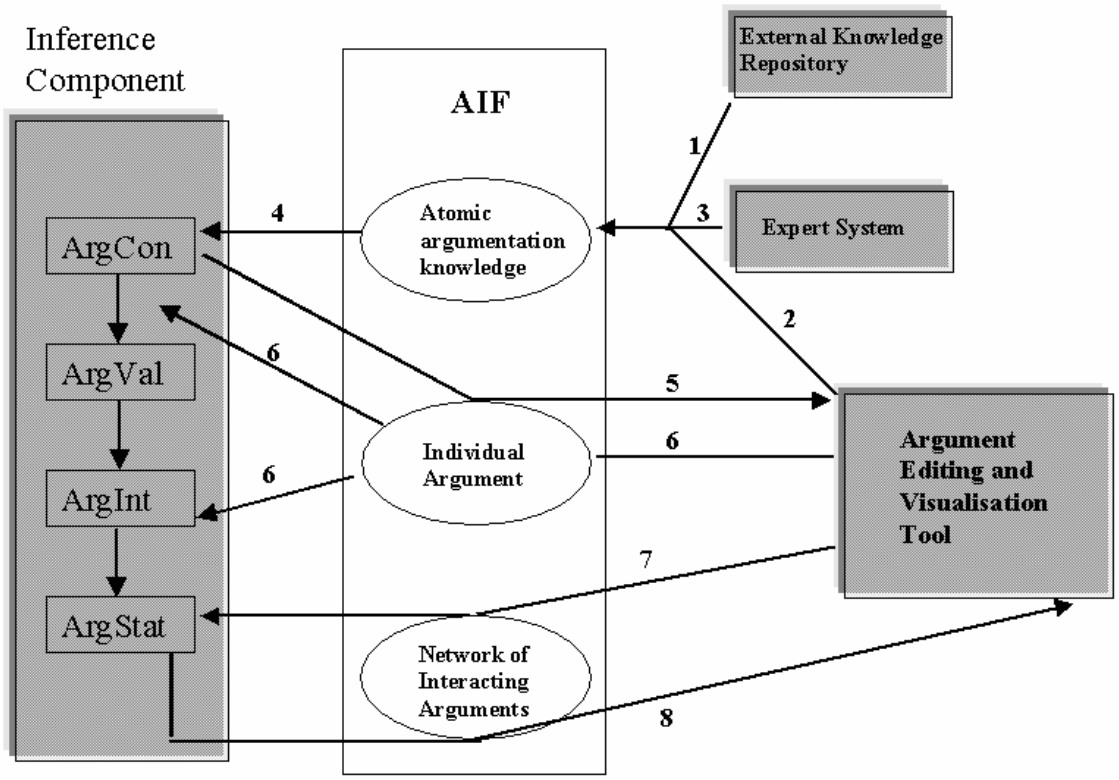


Figure 7 Linkage of inference component to the external knowledge and visualization tool, illustrating representational requirements for AIF

4. The majority of existing tools lack mechanisms for automated evaluation of the dialectical status of the interacting arguments authored by a user (e.g. none implements evaluation of the Dung acceptability of arguments).
5. Furthermore, consider that what gives visualization high utility in an argumentation context is that argumentation is inherently a process (Loui, 1998) rather than an instant, static picture. Users can enhance their interaction in this explicit and visual process by rendering visualization of the algorithmic proofs of acceptability of an argument. These proofs take the form of an argument game between a proponent and opponent (see, e.g., Cayrol *et al.* (2003)) that can be displayed visibly to a user as a tree (arrow 8) in which, as in the case of a Dung framework, links between nodes (arguments) represent attacks. Again, this would require mapping to some representation format that can be provided as an input to the visualization tool for rendering diagrammatically. Making transparent the workings of the inference component can enable the user to interact, adopting the role of proponent or opponent by submitting further arguments (as well as possibly information describing their strength and interactions with other arguments) as indicated by arrow 6.

5 Reifications

In this section we describe three example reifications of the AIF concepts defined in Section 3.2. By 'reification,' we mean a specific syntax, concretizing the AIF's abstract definitions, which can be unambiguously serialized and de-serialized for transmission between two communicating participants/software tools exchanging arguments. Note that in general the following apply.

1. More than one reification may exist.
2. Two different reifications may not be interoperable. That is, serializers for one reification may produce output which is not readable by parsers for another. The ontology specified here aims to make translations between reifications possible.

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press

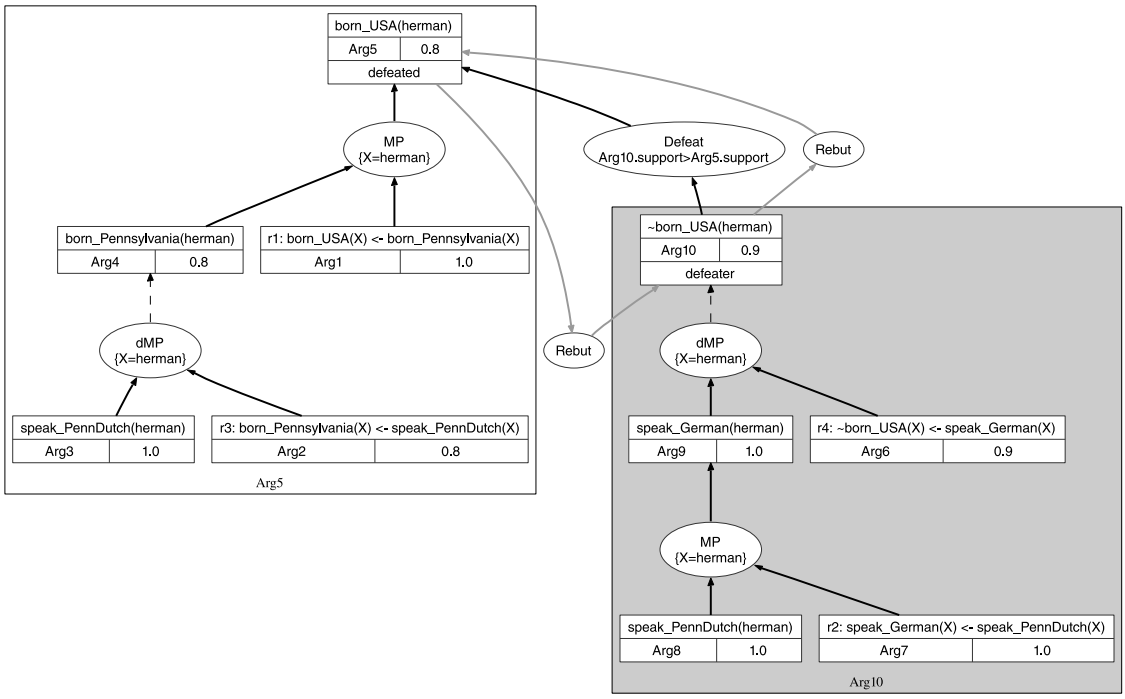


Figure 8 An example inference proof graph as generated by the ASPIC inference prototype. The proof graph is produced after reasoning over the claim '`is Herman born in the USA?`'. In this instance, using logical contradiction and defeat based on weakest link argument valuation, the claim is deemed inadmissible

3. While individual reifications will each aim to capture the semantics of the concepts defined in the AIF ontologies, they may also be influenced by the semantics of the encoding language used. Hence, minor semantic differences as well as syntactic differences may arise.

5.1 An example reification of the AIF in the ASPIC project

An important use case identified in Section 4.2 is the ability to communicate an evaluated AN that can then be unambiguously interpreted by the receiver. This use case also subsumes other use cases such as the ability to represent a single argument tree and argument interactions and has been implemented in the ASPIC prototype inference engine. The engine uses a PROLOG-like syntax for users to define strict and defeasible rules and beliefs and query those rules and beliefs using several variations of Dung semantics. The main outputs of the prototype engine are a human-readable diagram for the proof graph of the query and a machine consumable XML document that represents the same graph with some additional context.

An example proof diagram is shown in Figure 8. The proof shown is associated with the query, '`born_USA(herman)?`' and the following knowledge base:

```
born_USA(W)           <- born_Pennsylvania(W).
speak_German(X)       <- speak_PennDutch(X).
born_Pennsylvania(Y) <- speak_PennDutch(Y) 0.8.
~born_USA(Z)          <- speak_German(Z) 0.9.

speak_PennDutch(herman) .
```

In the prototype language, defeasibility is indicated with a weight (0, 1), known as 'degree of belief'. The weight allows an ordering over rules and facts that can be used to compare arguments. Thus, the first two rules in the example knowledge base are strict and the last two defeasible, with the fourth being preferred to the third. The query shown is evaluated using preferred credulous

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press

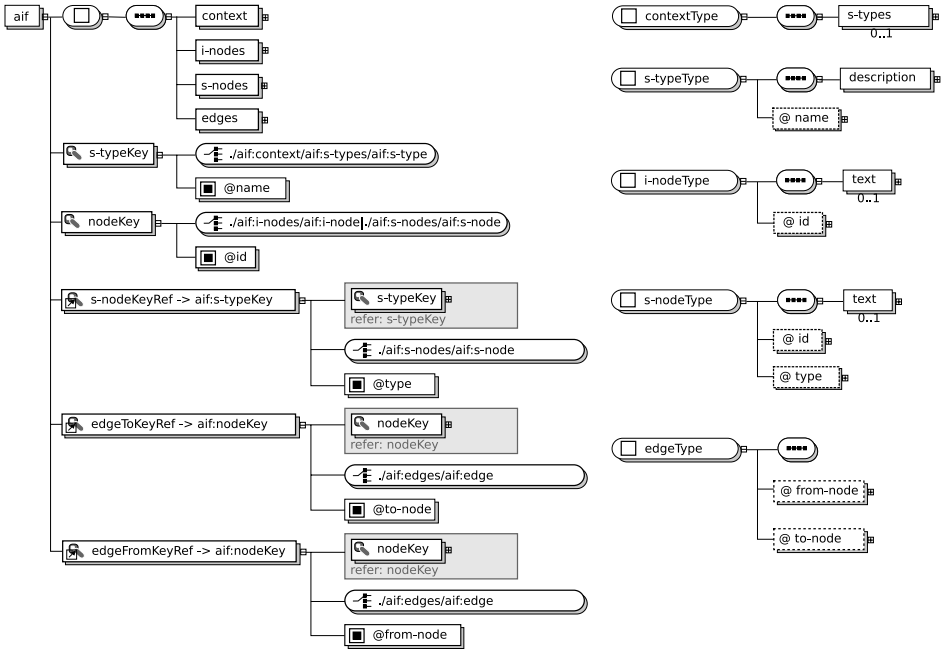


Figure 9 Diagrammatic view of the AIF XML schema as drafted for ASPIC. The top-level, *aif* element has four 'container' subelements: context, S-nodes, I-nodes and edges. These are shown along with unique key and keyref constraints in the left-hand side of the diagram. The right-hand side of the diagram indicates the design of the container subelements

semantics (Dung, 1995) and weakest link argument valuation. Other options include grounded semantics, last link valuation and restricted rebutting (Caminada & Amgoud, 2005) where an argument whose top rule is defeasible cannot rebut an argument whose top rule is strict.

Two XML schemas have been defined for ASPIC. The first is a general AIF schema that captures and constrains the minimal schema required for defining an AIF argument graph in XML. The second is a specific schema which extends the first schema with implementation details from the inference engine. A diagram showing the design of the general schema is shown in Figure 9. It is anticipated that the majority of AIF generators will be able to inherit this schema and extend it for their particular purposes, and this is what has been done for the case of the inference engine schema. One constraint that is not expressible in the current draft of the general schema is that I-nodes may not be directly connected to other I-nodes (via an edge). This constraint would have to be explicitly checked by consumers of this schema, and not captured by schema validation tools alone.

The particular schema for the inference engine extends the general schema in several ways. The context is expanded with details of the generating engine, the knowledgebase, the query options and the result of the query. I-nodes are extended with a type (rule or fact), a qualifier (degree of belief) and a status (defeated, defeater or admissible). The status attribute allows individual argument trees to be identified because only the top arguments in an argument tree are given a status by the engine. Finally, S-nodes are extended with details of the substitution used to form the grounded claim of an argument in inference scheme application nodes.

5.2 An example reification of the AIF in Araucaria

To demonstrate the breadth of the proposed AIF ontology, a second reification will be used to show how our proposal can accommodate a mature extant representation scheme, namely Araucaria's AML (Reed & Rowe, 2004). One of the advantages of exploring the relationship between AML and the AIF is that Araucaria and the other tools that support AML cater for a variety of different theoretical frameworks and approaches. If AML as a whole can be mapped into AIF, then by extension the same applies for these different approaches. We briefly consider here three styles

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press

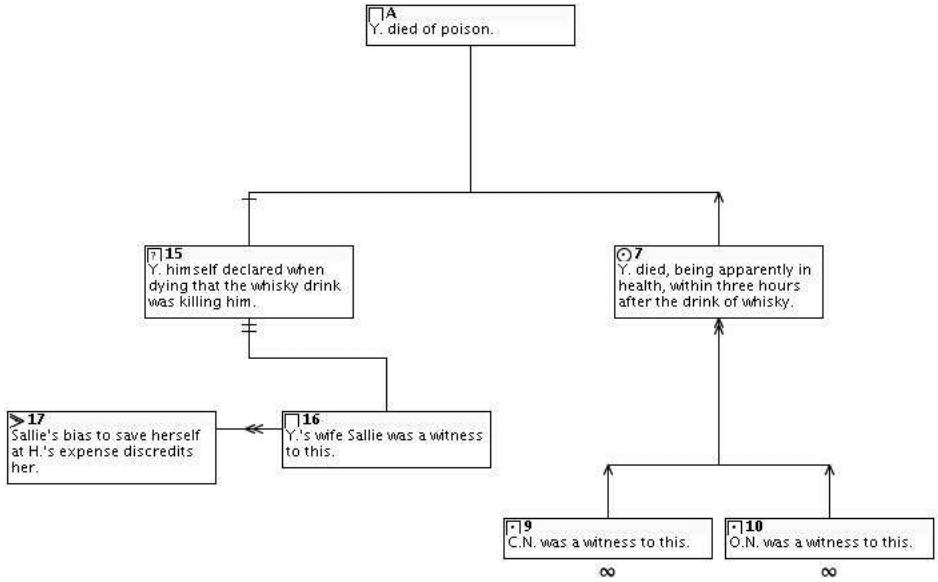


Figure 10 The poison case—Wigmore analysis

supported by Araucaria: (1) the traditional box-and-arrow diagrams that not only characterize formal diagrammatic approaches to argument theory, but that also represent an intuitive and straightforward visual language for back-of-the-envelope style jottings; (2) the influential Toulmin approach (see Section 3.3.2) with a strong jurisprudential and pedagogic heritage; (3) the Wigmore style analysis, used exclusively in legal practice (Wigmore, 1913). Although the three approaches differ significantly in their ontological characters, intended audiences and domains of use, they also share a number of features that has made them amenable to AML-based representation; this means they form just a subset of what the AIF can represent.

The example shown in Figure 10 is taken from one of Wigmore's cases, *Hatchett v. Com.*, and shows just a small part of the argument. In summary, the prosecution is trying to show that Y. died of poison (A). Two lines of argument are provided: the first is circumstantial, stating that Y. fell dead; the second is testimonial, based on what Y. said before dying (that the whisky was killing him) (15). The first claim has two witnesses (9, 10), the second just one (16). The defence attacks the second witness claim by pointing out the corroborative evidence that the witness is biased (17). The various other marks indicate the strength of relationships between the arguments, and that evidence has been admitted to the jury (the ∞ on boxes 9 and 10).

The same analysis when rendered in Araucaria's box-and-arrow approach is shown in Figure 11, which emphasizes the scheme relationships but does not show different classes of evidence. This approach also highlights that something interesting is going on with the defence's claim of Sallie's bias: it is working to attack the witness testimony inference scheme in an undercutting style.

Given the inter-translation between these two versions of the same analysis, we would hope that the AIF version would support either interpretation. As Figure 11 demonstrates, the AIF ontology handles the underlying structures easily. The probandum (A: Y. died of poison) is supported through two separate S-nodes that embody specific rule applications. The first makes the link from Y.'s actual death (7). This in turn is supported, through two applications of the witness testimony argumentation scheme, by two individual accounts of his death (9, 10). The other line of support for A is from the application of a rule that links from the premise (15) (that Y. himself declared that the whisky was killing him). This premise in turn is supported through an application of the witness testimony scheme which is founded upon testimony evidence (16). However, there is an application of a conflict definition that undermines (or, specifically, undercuts) the application of the witness testimony scheme, founded upon the bias of the witness (17). This offers a clear example of how

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press

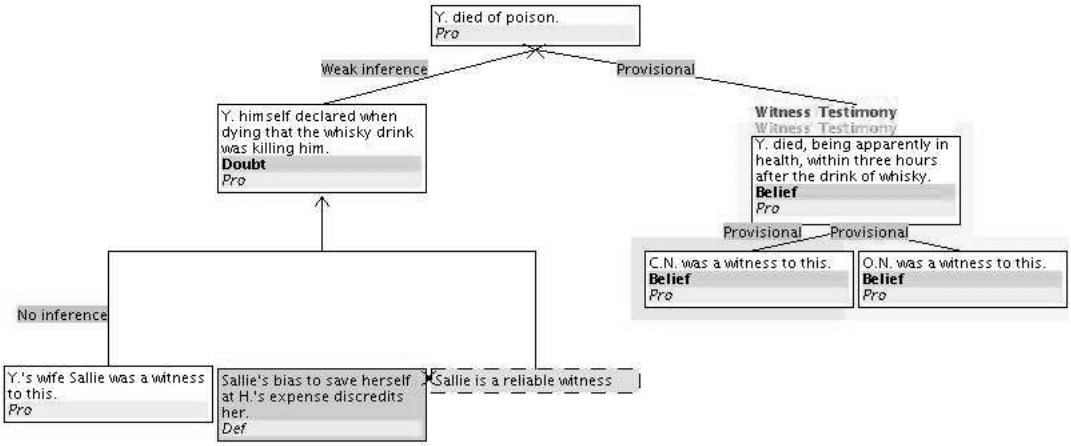


Figure 11 The poison case—standard analysis

conflict definitions function to encapsulate the critical questions associated with a particular presumptive argumentation scheme (Walton, 1996).

This AIF representation includes structural information, some of which is explicit in the box-and-arrow diagram, and some in the Wigmore analysis. The only information that is lost (such as the fine-grained analysis of evidence types provided in the Wigmore framework) can be reintroduced as values on node attributes in a Wigmore (or legalistic) extension to the core ontology.

5.3 An example reification of the AIF in RDF Schema/RDF

The reification of the AIF reported in this subsection is an attempt to exploit the potential of Semantic Web technology (Antoniou & van Harmelen, 2004) in argument representation and processing. The Semantic Web effort, led by Tim Berners-Lee, attempts to create a universal medium for information exchange by giving computer-understandable meaning (semantics) to the content of documents on the World Wide Web. Particularly useful for the context of the Semantic Web is the resource description framework (RDF)[13] which provides a general framework for describing Internet resources. The RDF defines a resource as any object that is uniquely identifiable by a uniform resource identifier (URI). Properties (or attributes) of resources are defined using an object–attribute–value triple, called a statement[14]. RDF statements can be represented as 3-tuples, as directed graphs or using a standard XML-based syntax. Unlike XML, which describes document models in directed-tree-based nesting of elements, the RDF model is based on arbitrary graphs, which are better suited for creating conceptual domain models. That way, the RDF can provide a more concise way of describing rich semantic information about resources. As a result, more efficient representation, querying and processing of domain models become possible.

RDF Schema (RDFS)[15] is an (ontology) language for specifying vocabularies in the RDF using terms described in the RDFS specification. RDFS provides mechanisms for describing characteristics of resources, such as the domains and ranges of properties, classes of resources or class taxonomies. RDFS (vocabulary specifying) statements are themselves described using RDF triples. Recently (Rahwan & Sakeer, 2006), we have first specified the AIF core ontology in RDFS using the Protégé ontology development environment[16]. For example, the following RDFS code declares the class `PA-Node` and states that it is a subclass of the class `S-Node`.

[13] `http://www.w3.org/RDF/`.
[14] Sometimes, an attribute is referred to as a property or a slot.
[15] `http://www.w3.org/TR/rdf-schema/`.
[16] `http://protege.stanford.edu/`.

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press

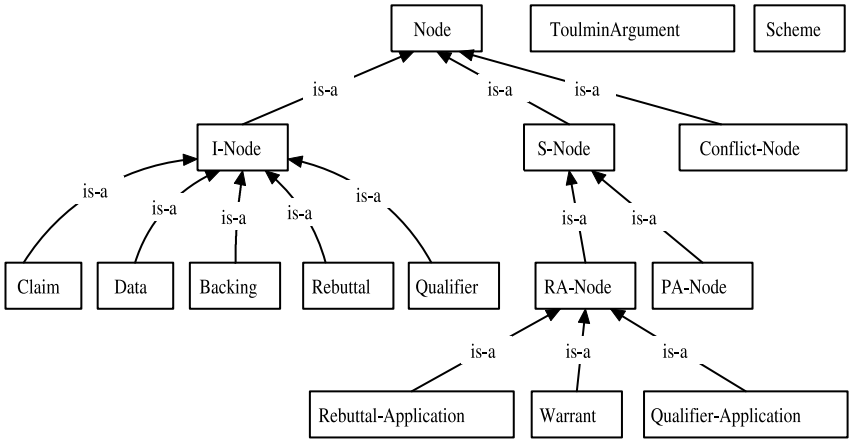


Figure 12 Toulmin argument class hierarchy in RDFS as an extension of the AIF ontology

```
<rdfs:Class rdf:about=''&kb;PA_Node'' rdfs:label=''PA_Node''>
  <rdfs:subClassOf rdf:resource=''&kb;S-Node''/>
</rdfs:Class>
```

Next, the following elements from the Toulmin scheme were introduced as I-Nodes: claim, data, backing, rebuttal and qualifier. All these elements represent passive declarative knowledge. Toulmin's warrant was expressed as an RA-Node, as it holds part of the argument together, namely the data nodes and the claim. Similarly, we introduced two other types of RA-Node: `Rebuttal-Application` nodes (used to link rebuttal nodes to claims) and `Qualifier-Application` nodes (used to link qualifier nodes to claims). The resulting ontology is represented in Figure 12.

Note that the concept `ToulminArgument` is a standalone concept. Instances of this concept will stand for complete arguments expressed in the Toulmin scheme. Such instances must therefore refer to instances of the various elements of the scheme. The ontology imposes a number of restrictions on these elements and their inter-relationships. In particular, each Toulmin argument must contain exactly one claim, exactly one warrant, exactly one qualifier, at least one backing and at least one data element. The following RDFS code declares the property `claim` which links instances of `ToulminArgument` to instances of type `Claim`, and states that each `ToulminArgument` must be linked to exactly one

```
<rdf:Property rdf:about=''&kb;claim">
  a:maxCardinality="1"
  a:minCardinality="1"
  rdfs:label=''claim''>
 <rdfs:domain rdf:resource=''&kb;ToulminArgument''/>
 <rdfs:range rdf:resource=''&kb;Claim''/>
<rdf:Property>
```

In our ontology, we defined various types of edges to capture every type of edge, such as those that emanate from backing nodes to warrant nodes, those from warrants to claims and so on. Note that according to our proposal a single claim node can belong to multiple instances of Toulmin arguments. Thus, for example, a single claim may be supported by multiple arguments. Moreover, a single data node could contribute to multiple unrelated claims. The RDF graph model enables such flexibility.

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press

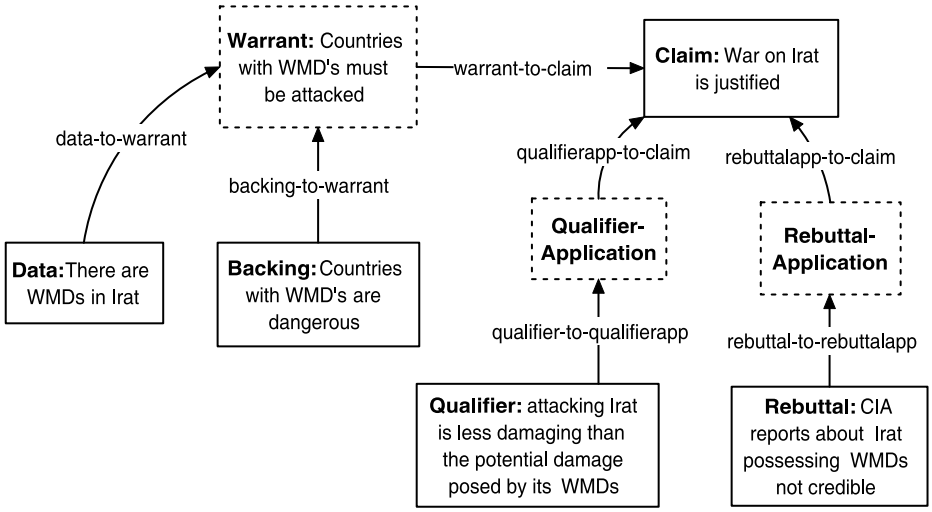


Figure 13 RDF graph for a Toulmin argument claiming that war on Irat is justified

With the ontology in place, it is now possible to create instances of the Toulmin argument scheme in RDF. Figure 13 shows an instance representing the argument mentioned above for justifying the war on an imaginary country called Irat.

In the figure, we have distinguished S-Nodes by dotted boxes although they are treated the same from the point of view of RDF processing.

Our ultimate aim is to provide an infrastructure for publishing semantically annotated arguments on the Semantic Web using a language that is semantically rich and amenable to machine processing. The choice of RDF as a representation language was motivated by its expressive power and the availability of tools for navigating and processing RDF statements.

In order to test our idea, we uploaded the argument instances on an installation of Sesame[17], an open source RDF repository with support for RDF Schema inferencing and querying. Sesame can be deployed on top of a variety of storage systems (relational databases, in-memory, filesystems, keyword indexers, etc.), and offers a large set of tools to developers to leverage the power of RDF and RDFS, such as a flexible access API, which supports both local and remote access, and several query languages, such as RQL and SeRQL. Sesame itself was deployed on the Apache Tomcat server, which is essentially a Java servlet container.

We have written a number of queries to demonstrate the applicability of our approach. The following query retrieves all warrants, data and backings for the different arguments in favour of the claim 'War on Irat justified'.

```
select WARRANT-TEXT, DATA-TEXT, BACKING-TEXT
from {WARRANT} kb:scheme-edge-warrant-to-claim {CLAIM},
  {WARRANT} kb:text {WARRANT-TEXT},
  {DATA} kb:data-edge-data-to-warrant {WARRANT},
  {DATA} kb:text {DATA-TEXT},
  {BACKING} kb:data-edge-backing-to-warrant {WARRANT},
  {BACKING} kb:text {BACKING-TEXT},
  {CLAIM} kb:text {CLAIM-TEXT}

where

  CLAIM-TEXT like ``War in Irat justified''
using namespace kb=http://protege.stanford.edu/kb#
```

[17] http://www.openrdf.org/.

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press

The output of the above query returned by Sesame will be the following:

WARRANT-TEXT	DATA-TEXT	BACKING-TEXT
Countries with WMD's must be attacked	There are WMD's in Irat	Countries with WMD's are dangerous

Query results can be retrieved via Sesame in XML for further processing. In this way, we could build a more comprehensive system for navigating argument structures through an interactive user interface that triggers such queries.

6 Conclusions and open issues

As discussed in Section 1, our AIF proposal aims at overcoming two major limitations present in currently available AMLs. On the one hand, existing AMLs are tailored to be used with a specific tool rather than for facilitating inter-operability of arguments among a variety of tools. On the other hand, these languages are primarily aimed at enabling users to structure arguments through diagrammatic linkage of natural language sentences rather than using formal logical statements. These limitations constitute a significant hindrance to the development and practical deployment of argumentation systems, mainly because of the lack of a shared, agreed notation or 'interchange format' for argumentation and arguments.

In order to solve this problem we have developed a draft specification for an AIF intended for representation and exchange of data between various argumentation tools and multi-agent reasoning and communication structures. The development of such an AIF is a highly challenging endeavour, as it requires providing an abstract model for capturing different features from a number of different scientific areas (such as argumentation theory, multi-agent systems and non-classical logics). In this context, it must be remarked that our proposal is a 'consensus' abstract model emerged from joint work among researchers from these different areas rather than a fully fledged proposal. Further, as noted in Section 3.2.5, the current model may well not capture all types of argumentation that are of interest. Specific significant open issues which arose during discussion included the following:

1. Currently, no distinction is being made for AIF formalisms which might be used in GUI/Tool import–export type application and those which might be used in agent-to-agent communication. While the core concepts may be the same it remains an open issue as to whether one format can really adequately cover both cases.
2. Given the potential richness of the communication concepts ontology, it remains an open issue as to how close to generic agent communication languages (ACLs; such as FIPA-ACL, KQML, etc.) AIF definitions may get. This affects possible re-use of ACL concepts and/or overlaps with them and/or worries about tractability issues concerning ACL semantics and consequently the semantics of the concepts defined here.
3. How should the community of users around the AIF organize themselves to agree on core concepts and extensions?
4. How should reifications be generated in detail from high-level concepts (e.g. development of specific RDF/XML schemas or other syntax forms)?

We think that future research for enhancing our current AIF proposal should aim at providing appropriate answers for some of these open issues. In this respect, contributions, suggestions and comments from other researchers interested in standardizing an AIF are welcome. Finally, it must be remarked that this paper is focused on presenting the final conclusions of our joint research work as a draft specification. A longer version of this document (containing initial inputs, previous versions and a discussion forum) for feedback can be found on the AIF Website at `http://x-opennet.org/aif/`.

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press

Acknowledgements

Support is gratefully acknowledged from Agentlink III[18] and the ASPIC project (FP6–IST-002307)[19] (both funded by the European Commission), as well as from Ramón y Cajal Programme (MCyT, Spain), from Spanish Projects TIC2003-00950, TIN2004-07933-C03-01/03 and TIN2004-07933-C03-03, and from CONICET (Argentina). Additional inputs and contributions are also gratefully acknowledged from all of the following: Leila Amgoud, Trevor Bench-Capon, Jamal Bentahar, Ivan Bratko, Martin Caminada, Sylvie Doutre, John Fox, Dan Grecu, David Hitchcock, Tsakou Ioanna, Paul Krause, Nicolas Maudet, Peter McBurney, Maxime Morge, Martin Mozina, Simon Parsons, Henri Prade, Henry Prakken, Glenn Rowe, Dave Robertson, Michael Rovatsos, Carles Sierra, Simon Wells and Michael Wooldridge.

While efforts have been made to reach a consensus on the content of this document, it is important to note that it remains the integration of a wide range of inputs, and hence the final result may not necessarily reflect the opinion of everybody who contributed—authorship or being listed as contributor does not necessarily imply complete agreement with the text.

References

Amgoud, L and Cayrol, C, 2002, A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence* **34**(1–3), 197–215.

Antoniou, G and van Harmelen, F, 2004, *A Semantic Web Primer (Cooperative Information Systems)*. Cambridge, MA: MIT Press.

Bench-Capon, TJM, 1997, Argument in artificial intelligence and law. *Artificial Intelligence and Law* **5**(4), 249–261.

Bench-Capon, TJM and Staniford, G, 1995, PLAID: proactive legal assistance. In *Proceedings of the 5th International Conference on Artificial Intelligence and Law*. New York: ACM Press, pp. 81–88.

Caminada, M and Amgoud, L, 2005, An axiomatic account of formal argumentation. In *Proceedings of the 20th National Conference on Artificial Intelligence and the 17th Annual Conference on Innovative Applications of Artificial intelligence*. Menlo Park, CA: AAAI Press.

Carbogim, D. *et al*., 2000, Argument-based applications to knowledge engineering. *Knowledge Engineering Review* **15**(2), 119–149.

Cayrol, C. *et al*., 2003, On decision problems related to the preferred semantics for argumentation frameworks. *Journal of Logic and Computation* **13**(3), 377–403.

Chesñevar, CI *et al*., 2000, Logical models of arguments. *ACM Computing Surveys* **32**(4), 337–383.

Chesñevar, C *et al*., 2006, Argument-based critics and recommenders: a qualitative perspective on user support systems. *Data & Knowledge Engineering* **59**(2), 293–319.

Conklin, J and Begeman, ML, 1988, gIBIS: a hypertext tool for exploratory policy discussion. *ACM Transactions on Office Information Systems* **6**(4), 303–331.

Dung, PM, 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.

Elhadad, M, 1995, Using argumentation in text generation. *Journal of Pragmatics* **24**, 189–220.

Emmet, L and Cleland, G, 2002, Graphical notations, narratives and persuasion: a pliant systems approach to hypertext tool design. In *HYPERTEXT 2002, Proceedings of the 13th ACM Conference on Hypertext and Hypermedia, 11–15 June 2002, University of Maryland, College Park, MD, USA*. New York: ACM Press, pp. 55–64.

FIPA, 2001, Communicative act library specification. Technical Report XC00037H, Foundation for Intelligent Physical Agents.

García, AJ and Simari, GR, 2004, Defeasible logic programming: an argumentative approach. *Theory and Practice of Logic Programming* **4**(1), 95–138.

Gordon, TF and Karacapilidis, N, 1997, The Zeno argumentation framework. In *Proceedings of the 6th International Conference on AI and Law*. New York: ACM Press, pp. 10–18.

Hamblin, CL, 1970, *Fallacies*. London: Methuen.

Kakas, A and Moraitis, P, 2003, Argumentation based decision making for autonomous agents. In *Proceeding of the 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'03)*, pp. 883–890.

[18] `http://www.agentlink.org`.
[19] `http://www.argumentation.org`.

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press

Kirschner, PA *et al.* (eds.), 2003, *Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making*. Berlin: Springer.

Loui, RP, 1998, Process and policy: resource-bounded nondemonstrative reasoning. *Computational Intelligence: An International Journal* **14**, 1–38.

Loui, R *et al.*, 1997, Progress on room 5: a testbed for public interactive semi-formal legal argumentation. In *Proceedings of the 6th International Conference on Artificial Intelligence and Law*. New York: ACM Press, pp. 207–214.

Maudet, N and Chaib-draa, B, 2002, Commitment-based and dialogue-game based protocols: new trends in agent communication languages. *Knowledge Engineering Review* **17**(2), 157–179.

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

McBurney, P and Parsons, S, 2003, Dialogue game protocols. In Huget, M-P (ed.), *Communication in Multiagent Systems (Lecture Notes in Computer Science, 2650)*. Berlin: Springer, pp. 269–283.

McBurney, P *et al.*, 2002, Desiderata for agent argumentation protocols. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002)*. New York: ACM Press, pp. 402–409.

McBurney, P *et al.*, 2005, The eight-fold way of deliberation dialogue. *Intelligent Systems*, in press.

Modgil, S, 2006, Hierarchical argumentation. In *Proceedings of the 10th European Conference on Logics in Artificial Intelligence, Liverpool, UK*.

Parsons, S *et al.* (eds.), 2006, *Argumentation in Multi-Agent Systems, 2nd International Workshop, ArgMAS 2005, Utrecht, the Netherlands, 26 July 2005, Revised Selected and Invited Papers (Lecture Notes in Artificial Intelligence, 4049)*. Berlin: Springer.

Pollock, J, 1987, Defeasible reasoning. *Cognitive Science* **11**, 481–518.

Prakken, H and Sartor, G, 1997, Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics* **7**, 25–75.

Prakken, H and Vreeswijk, GAW, 2002, Guest editorial: Logics for defeasible argumentation. In Gabbay, D and Günthner, F (eds.), *Handbook of Philosophical Logic*, vol. 4. Dordrecht: Kluwer Academic, pp. 219–318.

Rahwan, I, 2005, Argumentation in multi-agent systems. *Journal of Autonomous Agents and Multi-Agent Systems* **11**(2), 115–206.

Rahwan, I and Sakeer, PV, 2006, Towards representing and querying arguments on the Semantic Web. In Dunne, PE and Bench-Capon, TJM (eds.), *Computational Models of Argument (Proceedings of COMMA 2006)*. Amsterdam: IOS Press.

Rahwan, I *et al.*, 2003, Argumentation based negotiation. *Knowledge Engineering Review* **18**(4), 343–375.

Rahwan, I, 2005, Guest Editorial: Argumentation in multi-agent systems. *Journal of Autonomous Agents and Multi-Agent Systems* **11**(2), 115–125.

Reed, C, 2006, Representing dialogic argumentation. *Knowledge Based Systems* **19**(1), 22–31.

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

Reed, C and Rowe, GWA, 2004, Araucaria: software for argument analysis, diagramming and representation. *International Journal of AI Tools* **14**(3–4), 961–980.

Robertson, D, 2004, Multi-agent coordination as distributed logic programming. In *International Conference on Logic Programming, Sant-Malo, France*, pp. 416–430.

Shum, SB *et al.*, 2000, ScholOnto: an ontology-based digital library server for research documents and discourse. *International Journal of Digital Libraries* **3**(3), 237–248.

Shum, SB *et al.*, 2006, Modelling naturalistic argumentation in research literatures: representation and interaction design issues. *International Journal of Intelligent Systems, Special Issue on Computational Modelling of Naturalistic Argumentation*.

Sycara, K, 1992, The PERSUADER. In Shapiro, D (ed.), *The Encyclopedia of Artificial Intelligence*. New York: Wiley.

Toulmin, S, 1958, *The Uses of Argument*. Cambridge: Cambridge University Press.

Verheij, B, 1999, Automated argument assistance for lawyers. In *Proceedings of the 7th International Conference on Artificial Intelligence and Law*. New York: ACM Press, pp. 43–52.

Walton, D, 1996, *Argumentation Schemes for Presumptive Reasoning*. Hillsdale, NJ: Lawrence Erlbaum.

Walton, DN and Krabbe, ECW, 1995, *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. Albany, NY: SUNY Press.

Wigmore, JH, 1913, *The Principles of Judicial Proof*. Boston, MA: Little, Brown & Co.

https://doi.org/10.1017/S0269888906001044 Published online by Cambridge University Press