The Argument Interchange Format (AIF) Specification

Argumentation Research Group, School of Computing, University of Dundee

November 8, 2011

http://www.arg.dundee.ac.uk/aif

1 The Argument Interchange Format

The AIF is a communal project which aims to consolidate some of the defining work on computational argumentation [4]. Its aim is to facilitate a common vision and consensus on the concepts and technologies in the field so as to promote the research and development of new argumentation tools and techniques. In addition to practical aspirations, such as developing a way of interchanging data between tools for argument manipulation and visualization, a common core ontology for expressing argumentative information and relations is also developed. The purpose of this ontology is not to replace other (formal) languages for expressing argument but rather to serve as an abstract interlingua that acts as the centrepiece (Figure 1) to multiple individual argument languages such as, for example, the formal ASPIC$^+$ framework [8], [9]'s Description Logic formalisation and the format used by the Rationale argument visualization programme [1]. Note that direct translations between argumentation formats are optional as they are not needed if we have the AIF ontology as an interlingua.

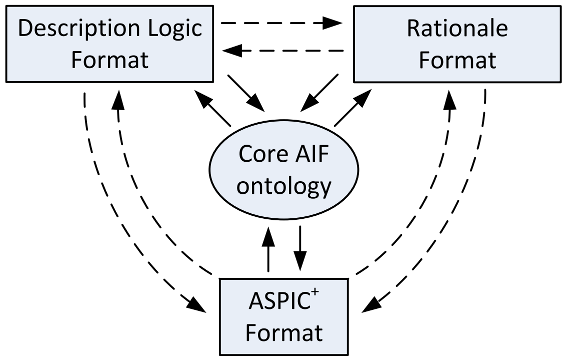


Figure 1: The AIF as the centrepiece of multiple argumentation languages

A common ontology for argumentation is interesting for a number of reasons. On the practical side, the AIF as an interlingua drastically reduces the number of translation functions that are needed for the different argumentation formats to engage with each other; only translation functions to the core AIF ontology have to be defined (i.e., n instead of n^2 functions for n argumentation formats). Furthermore, the central ontology acts as a conceptual anchoring point for the various formats, which improves the exchange of ideas between them. This anchoring point could provide a foundation for more formal characterisations of meaning within the different frameworks. By providing a strict graph-theoretic representation, the AIF provides a frame of reference accessible to theories of argument founded upon first order logics, higher order and non-classical logics, and theories of argument developed in epistemological contexts, linguistic contexts, and theories applied in pedagogy, law, and so on.

A common frame of reference, however, does not obviate the problem of commonness of meaning, for it still presupposes that the developers of the various argumentation theories have some sort of common understanding of the AIF core ontology. In order to promote this common understanding, the core ontology should be kept as basic as possible and the various elements of the ontology should be clearly defined. [10] note that due to the nature of the AIF project it is unavoidable that the ontology – and thus its common interpretation – will change over time. However, by having more translations and thus more references available the common understanding of the AIF will be further improved. Furthermore, the AIF project does not aim to tie applications or research projects to a particular format or interlingua. In some cases, for example when one wants two logical systems, it might be more sensible to provide a direct translation between the two systems which focuses on the formal properties of those systems (as is the case in, e.g., [16]). In a sense, the AIF ontology can be understood as a *tool* for development of interchanges because it can, as it were, provide a meeting point or forum for various researchers and application developers to define translation functions that facilitate interchange.

In this paper, we provide a specification of the AIF as used by the Argumentation Research Group at the University of Dundee. It is a complete and slightly more specific version of the ontology presented in [4, 11, 10]. The specification is available in various formats (OWL, SQL) from `http://www.arg.dundee.ac.uk/aif`.

1.1 The AIF Core Ontology

The AIF core ontology falls into two natural halves [11, 10]: the Upper Ontology and the Forms Ontology. The Upper Ontology defines the basic building blocks of AIF argument graphs, types of nodes and edges (in a sense, it defines the "syntax" for our abstract language). The Forms Ontology allows for the conceptual definition of the elements of AIF graphs, such as premises, inference schemes, exceptions and so on (it provides, for want of a better term, a "semantics" for the graph). Thus, the nodes defined in the Upper Ontology can be used to build argument-graphs at the object level (see Definition 1.1). The nodes in these graphs then *fulfil* (i.e. instantiate) specific argumentation-theoretic forms in the Forms Ontology.

Figure 2 visualises the main specification of the AIF ontology. The white nodes define the classes (concepts) in the Upper Ontology whilst the grey nodes define those in the Forms Ontology. Different types of arrows denote different types of relations between the classes in the ontology. For example, the class of *inference schemes* is a subclass of the class of *schemes*, an element of the class of RA-nodes fulfils an element of the class of *inference schemes* and elements of the class of *inference schemes* always have associated elements in the *premise* class. The full AIF specification includes further constraints on the construction of argument-graphs, that is, on the possible combinations of different nodes. These constraints are listed in Definition 1.1. Note that the graph in Figure 2 should not be confused with an AIF argument graph as defined in Definition 1.1 and shown in the rest of this paper (Figures 3 – 5): Figure 2 shows the structure of the ontology as a graph (i.e. a semantic network, a fairly common way to express such information) whilst the argument graphs in Figures 3 – 5 show actual arguments in the language of this ontology.

1.2 The Upper Ontology: building argument graphs

The AIF Upper Ontology places at its core a distinction between *information*, such as propositions and sentences, and *schemes*, general patterns of reasoning such as inference or conflict. Accordingly, the Upper Ontology describes two types of nodes for building argument graphs: information nodes (I-nodes) and scheme nodes (S-nodes). Scheme nodes can be rule application nodes (RA-nodes), which denote specific inference relations, conflict application nodes (CA-nodes), which denote specific conflict relations, and preference application nodes (PA-nodes), which denote specific preference relations. As [4] notes, nodes can have various attributes (e.g. `creator`, `date`). For current purposes, we assume that a node consists of some content (i.e. the information or the specific scheme that is being applied) and some identifier.

Nodes are used to build an *AIF argument graph* (called argument networks by [11, 10]). The choice of graphs as the AIF ontology's main representational language seems to be the most intuitive way of representing argument in a structured and systematic way without the formal constraints of a logic [4]. Furthermore, the graphical representation of arguments fits well with many textbook accounts of argument structure (see, for example, [5, 6, 17, 7]) and allows for the easy visualization of relations between nodes. However, the choice of the representational language is in some ways arbitrary and some AIF specifications [9] do not explicitly opt for a graph-based language.

An AIF argument graph can be defined as follows:

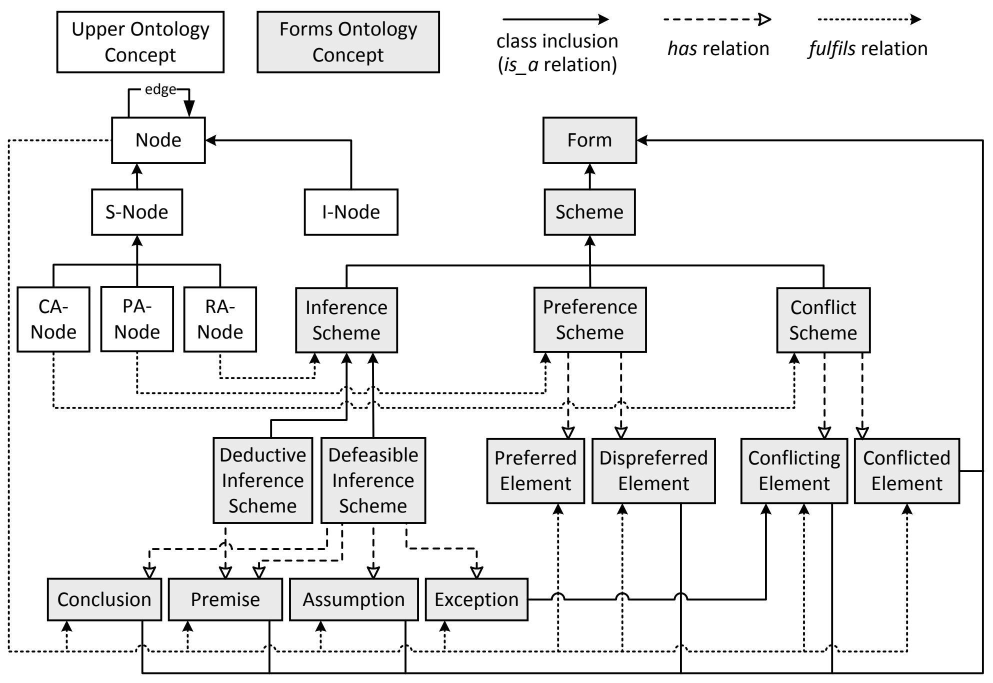


Figure 2: The AIF specification

Definition 1.1 [AIF graph]
An *AIF argument graph* G is a simple digraph (V, E) where

1. $V = I \cup RA \cup CA \cup PA$ is the set of nodes in G, where I are the I-nodes, RA are the RA-nodes, CA are the CA-nodes and PA are the PA-nodes; and

2. $E \subseteq V \times V \setminus I \times I$ is the set of the edges in G; and

3. if $v \in V \setminus I$ then v has at least one direct predecessor and one direct successor; and

4. if $v \in RA$ then v has at least one direct predecessor that fulfils the form *premise* and exactly one direct successor that fulfils the form *conclusion*; and

5. if $v \in PA$ then v has exactly one direct predecessor v_i that fulfils the form *preferred element* and exactly one direct successor v_j that fulfils the form *dispreferred element*, where $v_i \neq v_j$; and

6. if $v \in CA$ then v has exactly one direct predecessor that fulfils the form *conflicting element* and exactly one direct successor that fulfils the form *conflicted element*.

We say that, given two nodes $v_1, v_2 \in V$, v_1 is a *predecessor* of v_2 and v_2 is a *successor* of v_1 if there is a path in G from v_1 to v_2, and v_1 is a *direct predecessor* of v_2 and v_2 is a *direct successor* of v_1 if there is an edge $(v_1, v_2) \in E$. A node v is called an *initial node* if it has no predecessor.

Condition 2 states that I-nodes can only be connected to other I-nodes via S-nodes, that is, there must be a scheme that expresses the rationale behind the relation between I-nodes. S-nodes, on the other hand, can be connected to other S-nodes directly (see, e.g., Figures 4, 5). Condition 3 ensures that S-nodes always have at least one predecessor and successor, so that (a chain of) scheme applications always start and end with information in the form of an I-node.

Conditions 3 – 6 state specific constraints on the argument graph which are determined by the Forms Ontology (Section 1.3): inference applications (RA-nodes) always have at least one *premise* node and at most one *conclusion* node (4), preference applications are always between two distinct nodes of the forms *preferred element* and *dispreferred element* (5) and conflict applications always have exactly one *conflicting element* node and one *conflicted element* node.

In [4, 11], it is argued that edges in a graph need not be typed and that the meaning of edges can always be inferred when necessary from the types of nodes they connect. There are, however, some exceptional situations. For example, an edge between an RA-node and a PA-node can denote that the preference (PA-node) is the conclusion of the inference (RA-node), or it can denote the situation where the inference is preferred to another inference (when there is another edge from the PA-node to this other RA-node). In such a case, we need to know which forms are fulfilled by the nodes. That is, is the RA-node a *preferred element* w.r.t. the PA-node, or is the PA-node a *conclusion* w.r.t. the RA-node instead? In the following section, we will briefly discuss the Forms Ontology and how it solves such ambiguities.

1.3 The Forms Ontology: defining reasoning schemes

The Forms Ontology defines the various schemes and types of statements commonly used in argumentation. In this paper, we will leave the exact structure of the Forms ontology largely implicit and simply assume the Forms Ontology is a set $\mathcal{F}$ that contains the relevant forms and schemes. Nodes in the argument graph fulfil forms in $\mathcal{F}$. The AIF does not commit to any particular formalisation of how fulfilment works; instead, we simply state that a node v in an argument graph fulfils a form $f \in \mathcal{F}$.[1] The cornerstones of the Forms Ontology are schemes. In the AIF ontology, inference, conflict and preference are treated as genera of a more abstract class of schematic relationships [3], which allows the three types of relationship to be treated in more or less the same way, which in turn greatly simplifies the ontological machinery required for handling them. Thus, inference schemes, conflict schemes and preference schemes in the Forms Ontology embody the general principles expressing how it is that q is inferable from p, p is in conflict with q, and p is preferable to q, respectively. The individual RA-, CA- and PA-nodes that fulfil these schemes then capture the passage or the process of actually inferring q from p, conflicting p with q and preferring p to q, respectively.

1.3.1 Inference Schemes

Inference schemes in the AIF Forms ontology are similar to the rules of inference in a logic, in that they express the general principles that form the basis for actual inference. They can be deductive (e.g. the inference rules of propositional logic) or defeasible (e.g. [18]'s argumentation schemes) and accordingly, we assume that $\mathcal{F}$ contains separate subsets of deductive and defeasible inference schemes. As can be seen in Figure 2, both types of inference scheme have some *premises*, containing descriptions of the scheme's premises, and a *conclusion*, describing the scheme's conclusion. If desired, more elements of the scheme can be defined in the Forms Ontology, such as *presumptions* or *exceptions* for defeasible schemes [11].

One example of an inference scheme is Scheme for Argument from Expert Opinion [18]:

Scheme for Argument from Expert Opinion
premises: E asserts that A is true (false), E is an expert in domain D containing A;
conclusion: A is true (false);
presumptions: E is a credible expert, A is based on evidence;
exceptions: E is not reliable, A is not consistent with what other experts assert.

[1][11] schemes in the Forms Ontology as individuals and denote fulfilment through a "fulfil" relation between nodes in the graph and forms. [9] define schemes as combinations of classes of statements in OWL Description Logic (DL) and let the machinery of DL handle fulfilment.

An argument based on this scheme is rendered in Figure 3, in which the Forms that the nodes fulfil are indicated next to the nodes. The example implicitly assumes that self-esteem issues are are in the domain of psychology. Note also that the schemes exceptions and presumptions are not shown in the example: these come into play when the argument is attacked (see 4). Thus, specific (but still generalizable) knowledge can be modelled in the AIF in a principled way using argumentation schemes, for which we can assume, for example, a raft of implicit assumptions which may be taken to hold and exceptions which may be taken not to hold. Note that the AIF ontology itself does not legislate which schemes are in $\mathcal{F}$ and the exact structure of these schemes; rather, this depends on the inference rule schemes or argumentation schemes that a particular reification format uses.

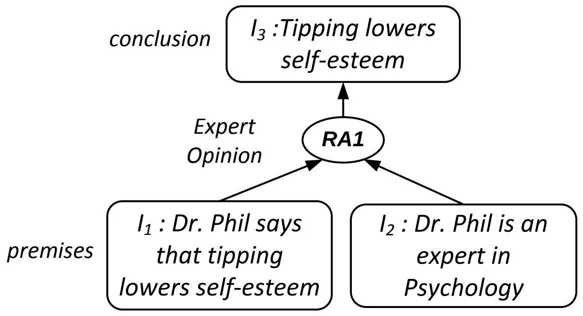

conclusion
I_3 :Tipping lowers self-esteem
Expert Opinion
RA1
premises
I_1 : Dr. Phil says that tipping lowers self-esteem
I_2 : Dr. Phil is an expert in Psychology

Figure 3: Argument from Expert Opinion

1.3.2 Conflict Schemes

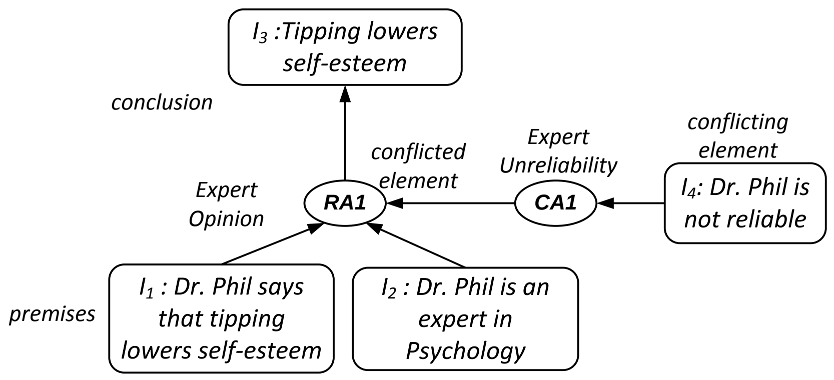
Like inference, conflict is also generalizable. General conflict relations, which may be based on logic but also on linguistic or legal conventions, can be expressed as *conflict schemes* in $\mathcal{F}$. All conflict schemes have two elements: one element that "conflicts" and another one that "is conflicted"; symmetry is not automatically assumed so that for a symmetrical conflict a scheme has to be applied twice.[2]

As an example of a conflict scheme, take the scheme for Conflict From Expert Unreliability, which states that that the fact that an expert is unreliable is in conflict with the inference based on the Expert Opinion scheme. In other words, the *conflicting element* of this scheme is 'E is not reliable' and the *conflicted element* is the Scheme for Argument from Expert Opinion. Note that (Figure 2) *exceptions* are also *conflicting elements*. This is also the case here: the exception of the Expert Opinion inference scheme (E is not reliable) is the conflicting element of the Expert Reliability conflict scheme. Figure 4 shows the application of the conflict scheme Expert Unreliability, which here attacks the application of the Expert Opinion inference scheme as represented by the node *ra12* (i.e. the fact that Dr. Phil is not reliable is in conflict with the fact that 'Tipping Lowers self-esteem' is inferred from the premises).

1.3.3 Preference Schemes

While inference and conflict allow us to build arguments and provide counterarguments, in many contexts a choice needs to be made as to which of the arguments is better or stronger. This can be expressed using preferences. In the AIF ontology, preference follows the now-familiar pattern that inference and conflict also follow: we assume a set of preference schemes

[2]Roughly, the *conflicting element* can be seen as the proposition that attacks and the *conflicted element* as the proposition that is attacked. However, because the notion of "attack" already has its own meaning in theories of computational argumentation, we are here forced to use the (rather cumbersome) terms *conflicting element* and *conflicted element*.

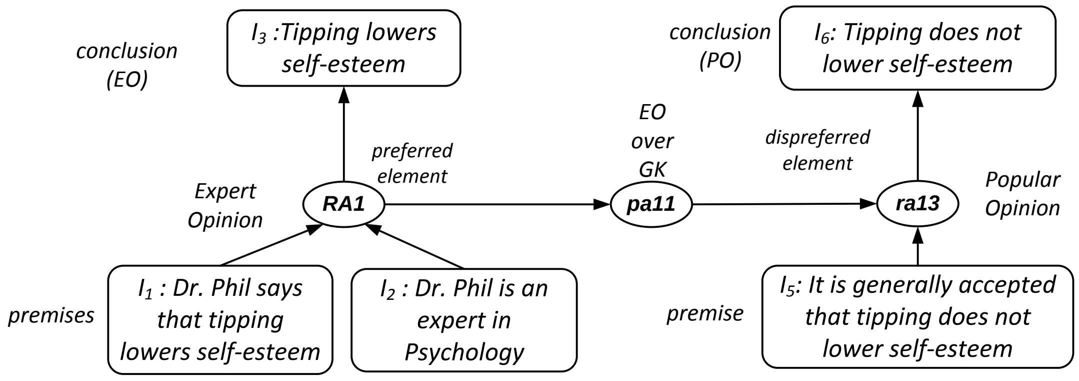


I_3 :Tipping lowers self-esteem
conclusion
conflicted element
Expert Unreliability
conflicting element
Expert Opinion
RA1
CA1
I_4: Dr. Phil is not reliable
premises
I_1 : Dr. Phil says that tipping lowers self-esteem
I_2 : Dr. Phil is an expert in Psychology

Figure 4: Attacking an Argument from Expert Opinion

in $\mathcal{F}$, which express principles for why certain (types of) information or schemes are preferable to others. A preference scheme contains a *preferred element*, the information or scheme that is preferred, and a *dispreferred element*, the information or scheme that the former is preferred to.

Preference schemes can define preferences between other schemes, for instance, inference schemes. As an example, consider an inference scheme for Popular Opinion [18], with as its premise 'A is generally accepted as true (false)' and its conclusion 'A is true (false)'. Now, we might want to say that, in general, arguments based on expert opinion are preferable to those based on popular opinion. This can be represented as a preference scheme with as its *preferred element* the inference scheme for Expert Opinion and as its *dispreferred element* the inference scheme for Popular Opinion. The preference scheme can then be applied as in Figure 5. Notice that because the scheme expresses that generally, inferences based on Expert Opinion are preferable over inferences based on Popular Opinion, then in this case the actual inference based on the Expert Opinion scheme (*ra12*) is preferred over the inference based on the Popular Opinion scheme (*ra13*). This example shows that a Forms Ontology is needed to disambiguate some of the elements of more complex graphs. For example, the table of informal semantics for edges in [11] says that an edge between an RA-node and a PA-node denotes 'inferring a conclusion in the form of a preference application'. However, in Figure 5, there is an edge from *ra12* to *pa11* which stands for something different, namely that *ra12* is an inference used in preference to some dispreferred element.



conclusion (EO)
I_3 :Tipping lowers self-esteem
conclusion (PO)
I_6: Tipping does not lower self-esteem
EO over GK
preferred element
dispreferred element
Expert Opinion
RA1
pa11
ra13
Popular Opinion
premises
I_1 : Dr. Phil says that tipping lowers self-esteem
I_2 : Dr. Phil is an expert in Psychology
premise
I_5: It is generally accepted that tipping does not lower self-esteem

Figure 5: Preference between inferences

2 Formats and reifications

The AIF ontology as presented above is available in two main reifications, and each reification is available in different formats. Each reification basically adheres to the abstract ontology set out in this document; the differences between reification lie in how they model the various concepts from the abstract ontology. The two main reifications currently available are the *Core AIF* and the *AIF in Description Logic*, which are both also discussed in [10].

2.1 AIF Core

This reification is closest to the specification given in this document. It is based on [4, 11] and used in [12, 14, 15, 2, 3]. This reification has precisely the concepts (i.e. classes) and relations between them as pictured in 2. Arguments (collections of nodes and edges) and specific schemes are represented by individual members of the different classes. Figure 6 shows the example rendered in this ontology: the boxes (ABoxes) represent the argument and the scheme, and for each box the class is given as underlined text.

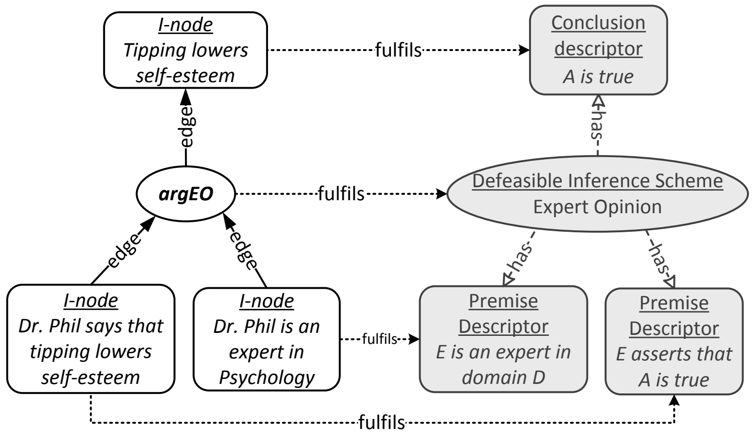


Figure 6: An argument in AIF Core

AIF Core is available in a number of standard ontology formats (RDF/XML, OWL/XML, Manchester OWL Syntax).[3]

2.1.1 AIF Database

AIF Core further forms the basis of the AIF-DB, a database containing a large amount of arguments analysed using the Araucaria[4] and the OVA[5] tools. The database structure, visualised in Figure 7, incorporates all the elements of the ontology in a standard SQL database: nodes, schemes and the links between them are simply elements of linked tables in the database. This database can be queried and the answer can then be output in various ways.

[3]www.arg.dundee.ac.uk/aif
[4]www.arg.dundee.ac.uk/araucaria, see [13]
[5]www.arg.dundee.ac.uk/ova

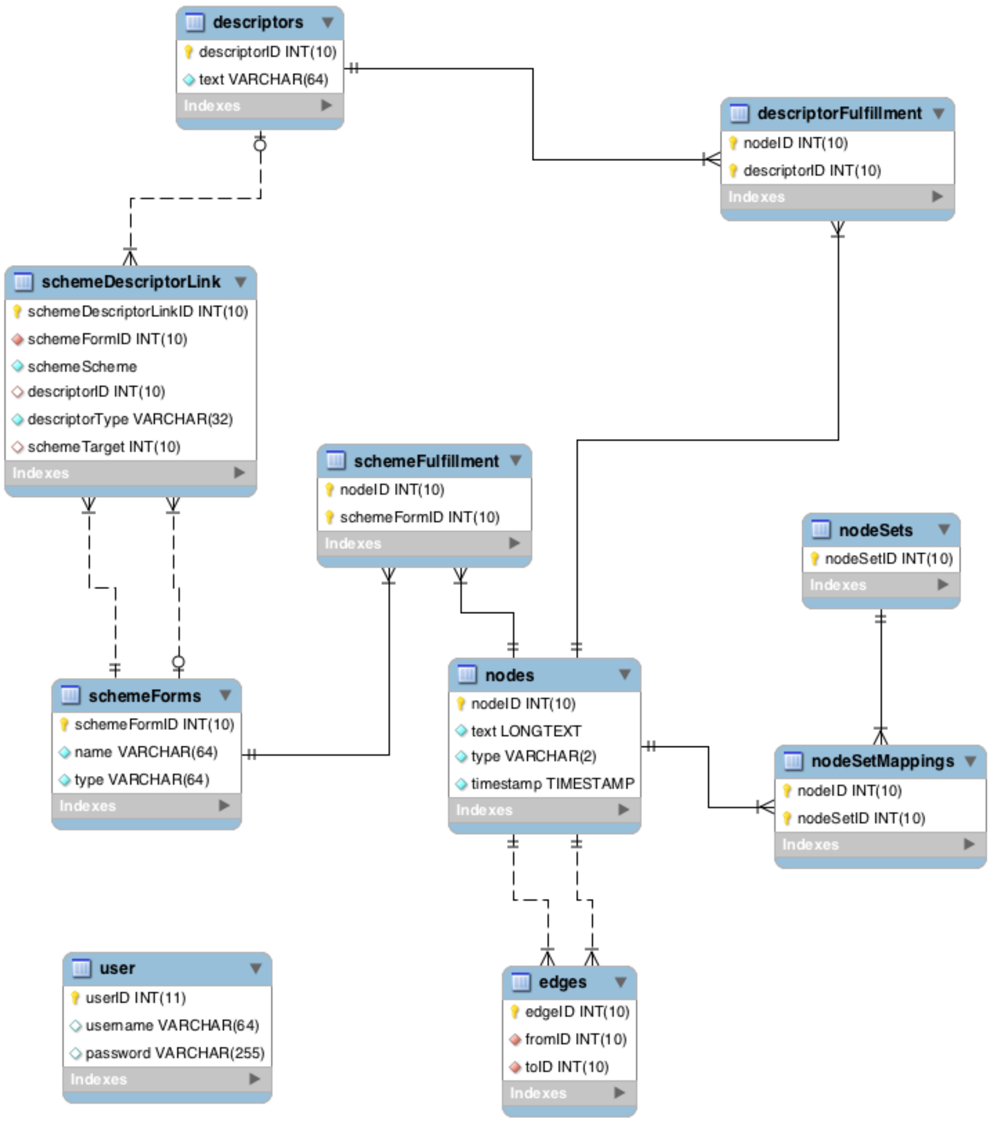


Figure 7: The AIF database

As an example, consider the argument in Figure 6. This argument, and its corresponding scheme, consists of elements of linked tables in the database, viz.:

Table 1: nodes

nodeID	text	type
1	Dr Phil says that tipping lowers self-esteem	I
2	Dr Phil is an expert on Psychology	I
3	Tipping lowers self-esteem	I
4	Dr Phil is not reliable	I
5	RA1	RA
6	CA1	CA

Table 2: edges

edgeID	fromID	toID
1	1	5
2	2	5
3	5	3
4	6	5
5	4	6

Table 3: schemeForms

schemeFormID	name	type
1	Expert Opinion	Defeasible Inference
2	Expert Unreliability	Conflict

Table 4: schemeFulfillment

nodeID	schemeFormID
5	1
6	2

Table 5: descriptors

descriptorID	text
1	E asserts that A is true
2	E is an expert in domain D
3	A is true
4	E is unreliable

Table 6: descriptorFulfill-ment

nodeID	descriptorID
1	1
2	2
3	3
4	4

Table 7: schemeDescriptorLink

ID	schemeFormID	schemeScheme	descriptorID	descriptorType	schemeTarget
1	1	FALSE	1	Premise	NULL
2	1	FALSE	2	Premise	NULL
3	1	FALSE	3	Conclusion	NULL
4	1	FALSE	4	Exception	NULL
5	2	FALSE	4	Conflicting Element	NULL
6	2	TRUE		Conflicted Element	1

2.2 AIF-DL

In [9], a slightly modified reification of the AIF Core ontology, formalised in Description Logic, was proposed: AIF-DL. AIF-DL captures the concept of schemes as *classes of arguments*, and S-nodes as instances of those classes. The rationale behind this modelling is that AIF Core's reification of the AIF causes some redundancy at the instance level: both arguments and schemes are described with explicit structure at the instance level (see Figure 6. As a result, the property "fulfils" does not capture the fact that an S-node represents an instantiation of some generic class of arguments. Having such relationship expressed explicitly can enable reasoning about the classification of schemes.

AIF-DB does not explicitly type ABoxes as I- and S-nodes. Instead, it distinguishes a class of statements (instances of which correspond to AIF Core I-nodes) and argument classes (i.e. schemes) that describe arguments made up of statements (instances of which correspond to AIF S-nodes). The "fulfils" relation is then simply captured by "instance of". Figure 6 shows the example rendered in this ontology: the boxes (ABoxes) represent the argument and the scheme, and for each box the class is given as underlined text.

AIF-DL is, like AIF-Core, available in a number of standard ontology formats (RDF/XML, OWL/XML, Manchester OWL Syntax).[6]

[6]`www.arg.dundee.ac.uk/aif`

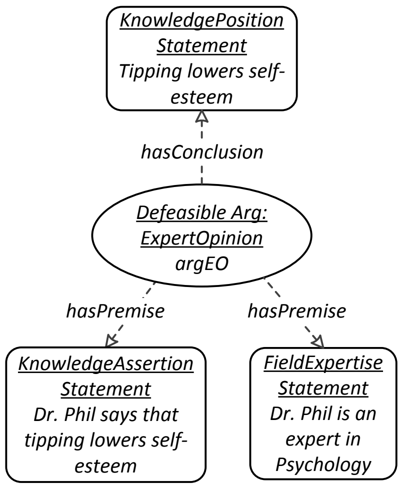

KnowledgePosition Statement
Tipping lowers self-esteem
hasConclusion
Defeasible Arg: ExpertOpinion
argEO
hasPremise
hasPremise
KnowledgeAssertion Statement
Dr. Phil says that tipping lowers self-esteem
FieldExpertise Statement
Dr. Phil is an expert in Psychology

Figure 8: An argument in AIF-DL

References

[1] T. Berg, T. van Gelder, F. Patterson, and S. Teppema. *Critical Thinking: Reasoning and Communicating with Rationale*. Amsterdam: Pearson Education Benelux, 2009.

[2] F.J. Bex, H. Prakken, and C.A. Reed. A formal analysis of the AIF in terms of the ASPIC framework. In P. Baroni, F. Cerutti, M. Giacomin, and G.R. Simari, editors, *Computational Models of Argument. Proceedings of COMMA 2010*, pages 99–110, Amsterdam, The Netherlands, 2010. IOS Press.

[3] F.J. Bex and C.A. Reed. Schemes of Inference, Conflict and Preference in a Computational Model of Argument. *Studies in Logic, Grammar and Rhetoric*, 2011.

[4] C.I. Chesñevar, J. McGinnis, S. Modgil, I. Rahwan, C. Reed, G. Simari, M. South, G. Vreeswijk, and S. Willmott. Towards an argument interchange format. *The Knowledge Engineering Review*, 21:293–316, 2006.

[5] J.B. Freeman. *Dialectics and the Macrostructure of Arguments: A Theory of Argument Structure*. Foris Publications, Berlin, 1991.

[6] T.F. Gordon, H. Prakken, and D.N. Walton. The Carneades model of argument and burden of proof. *Artificial Intelligence*, 171:875–896, 2007.

[7] J.L. Pollock. *Cognitive Carpentry: A Blueprint for How to Build a Person*. MIT Press, Cambridge, MA, 1995.

[8] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1:93–124, 2010.

[9] I. Rahwan, I Banihashemi, C. Reed, Walton D., and S. Abdallah. Representing and classifying arguments on the semantic web. *Knowledge Engineering Review*, 2010. Accepted for publication.

[10] I. Rahwan and C.A. Reed. The argument interchange format. In I. Rahwan and G. Simari, editors, *Argumentation in Artificial Intelligence*. Springer, 2009.

[11] I. Rahwan, F. Zablith, and C. Reed. Laying the foundations for a world wide argument web. *Artificial Intelligence*, 171:897–921, 2007.

[12] C. Reed, S. Wells, J. Devereux, and G. Rowe. Aif+: Dialogue in the argument interchange format. In Ph. Besnard, S. Doutre, and A. Hunter, editors, *Computational Models of Argument: Proceedings of COMMA-2008*, pages 311–323. IOS Press, 2008.

[13] C.A. Reed and G. Rowe. Araucaria: Software for Argument Analysis, Diagramming and Representation. *International Journal of AI Tools*, 13(4):961–980, 2004.

[14] C.A. Reed, S. Wells, K. Budzynska, and J. Devereux. Building arguments with argumentation : the role of illocutionary force in computational models of argument. In *Proceedings of COMMA 2010*, 2010.

[15] M. Snaith, J. Lawrence, and C. Reed. Pipelining argumentation technologies. In P. Baroni, F. Cerutti, M. Giacomin, and G.R. Simari, editors, *Computational Models of Argument. Proceedings of COMMA 2010*, pages 447–454, Amsterdam, The Netherlands, 2010. IOS Press.

[16] B. van Gijzel and H. Prakken. Relating Carneades with abstract argumentation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 1013–1019, 2011.

[17] B. Verheij. Deflog: on the logical interpretation of prima facie justified assumptions. *Journal of Logic and Computation*, 13:319–346, 2003.

[18] D.N. Walton, C.A. Reed, and F. Macagno. *Argumentation Schemes*. Cambridge University Press, Cambridge, 2008.