Laying the Foundations for a World Wide Argument Web

Iyad Rahwan [a,b,*] Fouad Zablith [a] Chris Reed [c]

[a] *Institute of Informatics, British University in Dubai, UAE*

[b] *(Fellow) School of Informatics, University of Edinburgh, UK*

[c] *School of Computing, University of Dundee, UK*

Abstract

This paper lays theoretical and software foundations for a *World Wide Argument Web (WWAW)*: a large-scale Web of inter-connected arguments posted by individuals to express their opinions in a structured manner. First, we extend the recently proposed Argument Interchange Format (AIF) to express arguments with a structure based on Walton's theory of argumentation schemes. Then, we describe an implementation of this ontology using the RDF Schema Semantic Web-based ontology language, and demonstrate how our ontology enables the representation of networks of arguments on the Semantic Web. Finally, we present a pilot Semantic Web-based system, ArgDF, through which users can create arguments using different argumentation schemes and can query arguments using a Semantic Web query language. Manipulation of existing arguments is also handled in ArgDF: users can attack or support parts of existing arguments, or use existing parts of an argument in the creation of new arguments. ArgDF also enables users to create new argumentation schemes. As such, ArgDF is an open platform not only for representing arguments, but also for building interlinked and dynamic argument networks on the Semantic Web. This initial public-domain tool is intended to seed a variety of future applications for authoring, linking, navigating, searching, and evaluating arguments on the Web.

Key words: Argument schemes, Argumentation, Tools, E-Democracy, Argument Interchange Format, Semantic Web

[*] Corresponding author
Email addresses: `irahwan@acm.org` (Iyad Rahwan), `fouad@zablith.org` (Fouad Zablith), `chris.reed@computing.dundee.ac.uk` (Chris Reed).
URLs: `http://homepages.inf.ed.ac.uk/irahwan/` (Iyad Rahwan), `http://fouad.zablith.org` (Fouad Zablith), `http://www.computing.dundee.ac.uk/staff/creed/` (Chris Reed).

1 Introduction

Argumentation can be defined as a verbal and social activity of reason aimed at increasing (or decreasing) the acceptability of a controversial standpoint for the listener or reader, by putting forward a constellation of propositions (i.e. arguments) intended to justify (or refute) the standpoint before a rational judge [53, page 5]. The theory of argumentation is a rich, interdisciplinary area of research encompassing but not exclusive to philosophy, communication studies, linguistics, and psychology.

A variety of opinions and arguments are presented every day on the Web, in discussion forums, blogs, [1] news sites, etc. As such, the Web acts as an enabler of large-scale argumentation, where different views are presented, challenged, and evaluated by contributors and readers. However, these methods do not capture the explicit structure of argumentative viewpoints. This makes the task of evaluating, comparing and identifying the relationships among arguments difficult.

First, let us outline our long-term vision through a scenario. You query the Web (e.g. through an appropriate form that generates a formal query) by asking a question like '*List all arguments that support the War on Iraq on the basis of expert assessment that Iraq has Weapons of Mass Destruction (WMDs).*' You are presented with various arguments ordered by strength (calculated using the number and quality of its supporting and attacking arguments). One of these arguments is a blog entry, with a semantic link to a CIA report claiming the presence of WMDs. You inspect the counterarguments to the CIA reports and find an argument that *attacks* them by stating that '*CIA experts are biased.*' You inspect this attacking argument and you find a link to a BBC article discussing various historical examples of the CIA's alignment with government policies, and so on.

Motivated by the above vision, we lay theoretical and software foundations of a *World Wide Argument Web (WWAW)*: a large-scale Web of inter-connected arguments posted by individuals on the World Wide Web in a structured manner. The theoretical foundation is an ontology of arguments, extending the recently proposed Argument Interchange Format [11], and capturing Walton's general theoretical account of argumentation schemes [57]. For the software foundation, we build on the strengths and potential of the Semantic Web [4] and implement the ontology using the RDF Schema Semantic Web ontology language. We then present a pilot Semantic Web-based system, ArgDF, through which users can create arguments using different argumenta-

[1] A *blog* (short for *Web-log*) is a user-generated website where entries (e.g. commentaries, news, diary items) are presented in journal style and displayed in a reverse chronological order.

tion schemes and can query arguments using a Semantic Web query language. Manipulation of existing arguments is also handled in ArgDF: users can attack or support parts of existing arguments, or use existing parts of an argument in the creation of new arguments. ArgDF also offers flexible features, such as the ability to create new argumentation schemes from the user interface. As such, ArgDF is an open platform not only for representing arguments, but also for building interlinked and dynamic argument networks on the Semantic Web. This initial public-domain tool is intended to seed what it is hoped will become a rich suite of sophisticated applications for authoring, linking, navigating, searching, and evaluating arguments on the Web.

This paper advances the state of the art in computational modelling of argumentation in three ways. First, it presents the first Semantic Web-based system for argument annotation, navigation and manipulation. Second, the paper provides the first highly scalable yet highly-structured argument representation capability on the Web. This contrasts with current group argumentation support systems, which are either scalable but weakly-structured, or highly-structured but theory-dependent and only applicable to small numbers of participants. Finally, the paper contributes to the recently proposed Argument Interchange Format (AIF) ontology [11] by extending it to capture Walton's argument schemes [57] and providing a complete implementation of the AIF in a Semantic Web language.[2] If successful, the WWAW will be the largest argumentation support system ever built because its construction is not centralised, but distributed across contributors and software developers in the model of many emerging Web 2.0 applications.[3]

The rest of the paper is organised as follows. In the next section, we discuss the different enabling components of large-scale argumentation. In Section 3, we present an overview of the current state of the Argument Interchange Format. We present our extensions to the AIF in Section 4 and discuss its RDFS implementation in Section 5. We then present the pilot system ArgDF in Section 6. We conclude the paper and discuss future potential applications in Section 7.

[2] To our knowledge, the only other representation of the AIF using Semantic Web languages is a preliminary attempt by the first author [40].

[3] *Web 2.0* is a term that has become widely used to refer to second-generation Web services that emphasise user collaboration, such as social networking sites, collaborative tagging sites (for so called *folksonomy* meta-data generation), mass collaborative editing (through wikis [28]), etc.

2 Enablers of Large-Scale Argumentation

Argumentation-based techniques and results have found a wide range of applications in both theoretical and practical branches of artificial intelligence and computer science [43] ranging from non-monotonic reasoning [37,10] to knowledge engineering [8], to multi-agent systems' communication and negotiation [38,39]. Another area that has witnessed significant growth is argumentation-support systems [25]. Our interest here is mainly in the latter, and particularly in large-scale argumentation support in a Web environment. By argumentation support, we mean tools that enable users to browse, visualise, search, and manipulate arguments and argument structures. There is a great diversity of resources that can be drawn upon in trying to build the foundation for the WWAW, including tools for interaction and visualisation, and, first and foremost, arguments themselves.

2.1 Arguments

The first important component of large-scale argumentation are the arguments themselves. In this sub-section, we discuss the availability of argument corpora, which may be used as a basis for providing argument search and navigation capabilities.

Currently, the largest corpus of analysed arguments is the *AraucariaDB* corpus from the University of Dundee [41]. It has around 500 arguments, produced by expert analysts, and drawn from newspapers, magazines, judicial reports, parliamentary records and online discussion groups from various countries and in different domains. Another significant analysis effort has been carried out at McMaster [22], and takes a smaller set of academic arguments as a sample upon which to evaluate aspects of theories of argument. *Globalargument*[4] is taking a different approach – that of encouraging many research groups to apply different analysis techniques to a common body of arguments. At the time of writing, the Globalargument community has managed several very detailed analyses of a single extended argument. Apart from these, no other academic effort at systematic analysis of arguments is known. Investigations such as those by Snoeck Henkemans [47,48] make use of an informal, closed corpus collected in Amsterdam. Salminen et al [46] describe a small-scale collection of specialised verbal arguments analysed in the context of the SCALE project. Argumentation theory as a field often makes use of small extracts to motivate techniques and conclusions [53]. But none of these represent the systematic collection of material to form a coherent corpus.

[4] See http://www.globalargument.net

The fact that argument analysis is difficult, slow and often disputable means that manual labour cost is high, which severely limits the scope of analysed and annotated arguments. Moreover, the approach relies on analyses by experts, which is also limiting. Finally, none of the above argument resources provides explicit links between the components of different arguments. They mainly focus on the analysis of a single argument at a time. This makes the process of navigating and searching interconnected argument within these corpora impossible.

One solution is to devolve the process of creating rich semantic models of arguments to the users of those arguments – rather than taking textual (or in some few cases, verbal) arguments as input to some centralised analysis process, instead facilitate analysis anywhere, by end users, or better still, encourage the creation of the semantically rich representations in the first place, avoiding the need for analysis entirely. This requires rich sets of tools – some generic, some tailored to specific domains; some focusing on analysis, some on rich generation. This, then, is the second set of extant resources: tools.

2.2 Tools for Arguing on the Web

The World Wide Web can be seen as an ideal platform for enhancing argumentative expression and communication, due to its ubiquity and openness. Personal blogs and unstructured or semi-structured on-line discussion forums can provide a medium for such communication. *Deme* [14] is an example of such a system, designed specifically for supporting democratic, small to medium-sized group deliberation. This approach, however, does not capture much of the structural attributes of the arguments under discussion. While opinions and discussions may be identified by their topics, time, or participants, there is a lack of fine-grained structure that captures how different facts, opinions, and arguments relate to one another and, as such, contribute to the overall picture. Having such structure has the potential to enable far better visualisation, navigation and analysis of the 'state of the debate' by participants or automated tools. Indeed, it has been shown that adding structure to on-line discussion environments improves the group's ability to reach consensus and make higher-quality decisions [16]. Moreover, such structure could make it easier to automate support for the argumentation process, for example, by discovering inconsistencies among arguments or by discovering synergies among disputants.

Recently, some Web-based tools have begun to enable simple structuring of arguments. The public argumentation support system *truthmapping*[5] supports a large number of participants but has very shallow structure. It only

[5] See `http://www.truthmapping.com`

distinguishes premises and conclusions, without providing a distinction among different types of arguments, and without cross-referencing complex interactions among arguments. A similar effort is being explored in *Discourse DB*, which was released to the public in late 2006. Discourse DB is a forum for journalists and commentators to post their opinions about ongoing political events and issues. [6] Opinions or arguments are organised by topic, and classified into three categories: for, against, and mixed. Moreover, content may be browsed by topic, author, or publication type. Discourse DB is powered by *Semantic MediaWiki* [56], which enables it to export content into RDF format for use by other Semantic Web applications.

A number of highly-structured argument-based deliberation support systems (ADSS) have been proposed. These systems suffer from two key limitations. Firstly, they usually support a small number of participants. Secondly, most of them target specific domains, such as education (e.g. Araucaria [45]), jurisprudence (e.g. ArguMed [54]), and academic research (e.g. ClaiMaker [51]). Consequently, they are based on specialised approaches of interaction and decision-making, rather than a general theory of argumentation. For example, Parmenides [2] is based on a specific inference scheme for justifying the adoption of an action, and a fixed set of possible attacks that can be made. Other ADSSs include *gIBIS* [13], *QuestMap*$^{\text{TM}}$[12], *SIBYL* [27], *Zeno* [19], *DEMOS* [29], *HERMES* [23], and *Risk Agora* [30,31].

Existing approaches to group argumentation and deliberation support suffer from a number of limitations. Firstly, there is a trade-off between scalability and structure. On one hand, scalable discourse support systems, such as discussion forums, Wikis and Blogs, lack the structure and argumentative rigour that most ADSSs offer. On the other hand, highly-structured ADSSs are based on client-server architectures and usually designed for small to medium-sized groups, and are therefore not easily scalable [18].

Another limitation of existing structured ADSSs is that they subscribe to specific theories of argumentation and decision-making. For example, the Parmenides system is based on a specific theory of persuasion over action. HERMES is based on elements such as issues, alternatives, positions, constraints and preferences. While these systems may be suitable for specific domains, a truly global-scale argumentation infrastructure must allow for a variety of reasoning patterns to structure interaction. Such reasoning patterns are known in argumentation theory as argumentation schemes [57].

Broadly speaking, current argumentation support technologies seem to present a trade-off. Large-scale discourse systems do not have enough structure to enable us to build powerful tools to support the visualisation, search, navigation

[6] See `http://discoursedb.org`

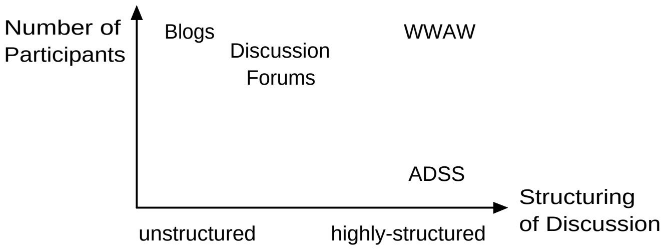


Fig. 1. Different approaches of argumentation support tools

and analysis of arguments by participants or automated tools, while highly-structured ADSSs are too restrictive in terms of scalability and the underlying reasoning patterns. To address this limitation, we need a theoretical and technological leap that achieves a global argumentation infrastructure that is highly scalable, yet highly customisable and structured (See Figure 1 for an illustration).

2.3 Desiderata

We propose a radically different approach to promoting large-scale argumentation. Instead of building yet another system for supporting discourse among small or medium-sized groups of participants, we aim to build an open, extensible and re-usable infrastructure for large-scale argument representation, manipulation, and evaluation.

In light of the above discussion, we now list a set of key requirements that we believe are important in order to allow for large-scale argument annotation on the Web.

(1) The WWAW must support the storage, creation, update and querying of argumentative structures;
(2) The WWAW must have Web-accessible repositories;
(3) The WWAW language must be based on open standards, enabling collaborative development of new tools;
(4) The WWAW must employ a unified, extensible argumentation ontology; and
(5) The WWAW must support the representation, annotation and creation of arguments using a variety of argumentation schemes;

In the next section, we outline the AIF core ontology originally reported by Chesñevar et al [11]. Our extensions to this core ontology (Section 4) form a basis for building the first prototype of the WWAW infrastructure (Sections 5 and 6).

3 Background: The Argument Interchange Format (AIF) Core Ontology

In this section, we outline the current state of the Argument Interchange Format (AIF), originally reported by Chesñevar et al [11]. We use a formal syntax in describing the elements of the AIF to simplify subsequent exposition.

The AIF is a core ontology of argument-related concepts. This core ontology is specified in a way that it can be extended to capture a variety of argumentation formalisms and schemes. To maintain generality, the AIF core ontology assumes that argument entities can be represented as nodes in a directed graph (also known as di-graph). This di-graph is informally called an *argument network.*

Arguments are represented using a set $\mathcal{N}$ of nodes connected by binary directed edges (henceforth referred to as *edges*) which we define using the predicate $\xrightarrow{edge}: \mathcal{N} \times \mathcal{N}$. We will sometimes write $n_1 \xrightarrow{edge} n_2$ to denote $(n_1, n_2) \in \xrightarrow{edge}$. A node can also have a number of internal attributes, denoting things such as textual details, or a numerical value indicating certainty degree or acceptability status, etc. Figure 2 visualises, through a semantic network [49], the classes of the AIF ontology and their interrelationships.

In this paper, in the interest of simplicity, we shall use a set-theoretic approach to describing the AIF. We will therefore use a *set* to define each *class* (or *type*) of things like nodes. So, the set $\mathcal{N}$ should be understood to denote the class of all nodes. And a particular sub-class $\mathcal{N}'$ of nodes will be captured as a subset of $\mathcal{N}$. An element $n \in \mathcal{N}$ is to be understood as an *instance* of that class, i.e. a particular node of type $\mathcal{N}$. This approach is similar to the way formal semantics are defined for Description Logics [3], which form the foundation for Semantic Web ontology languages such as OWL [33]. Finally, properties and relations between classes and instances (including graph edges) will be captured through predicates over sets.

There are two types of nodes in the core AIF: *information nodes* (or *I-nodes*) which hold pieces of information or data, and *scheme nodes* (or *S-nodes*) representing the inferential passage associated with an argumentative statement. These are represented by two disjoint sets, $\mathcal{N}_I \subset \mathcal{N}$ and $\mathcal{N}_S \subset \mathcal{N}$, respectively. We describe the nodes briefly below.

Information nodes are used to represent *passive* information contained in an argument, such as a claim, premise, data, etc. On the other hand, S-nodes capture the application of *schemes* (i.e. patterns of reasoning). Such schemes may be domain-independent patterns of reasoning, which resemble rules of inference in deductive logics but broadened to include non-deductive logics

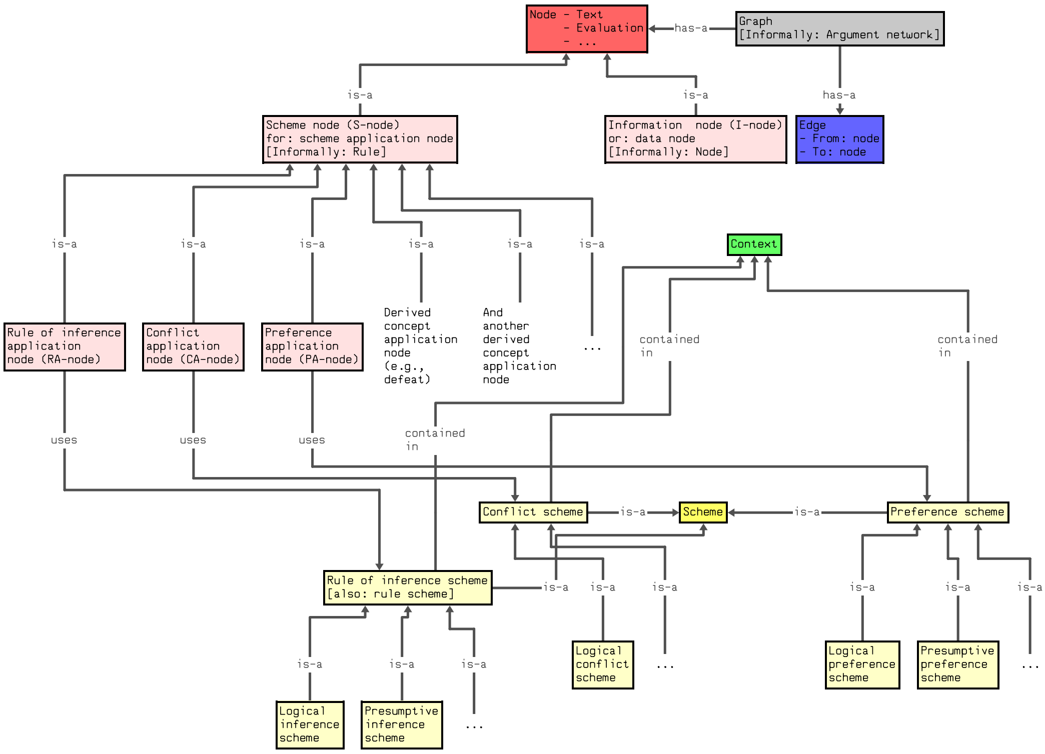


Fig. 2. Semantic network of concepts and relations in the AIF core ontology [11]

that are not restricted to classical logical inference. The schemes themselves belong to a class, $\mathcal{S}$, which are classified into the types: *rule of inference schemes*, *conflict schemes*, and *preference schemes*. We denote these using the disjoint sets $\mathcal{S}^R$, $\mathcal{S}^C$ and $\mathcal{S}^P$, respectively. The predicate $(\mathsf{uses} : \mathcal{N}_S \times \mathcal{S})$ is used to express the fact that a particular scheme node uses (or instantiates) a particular scheme. For example, we would require that each conflict application node is linked to a particular conflict scheme that it uses. The AIF thus provides an ontology for expressing schemes and instances of schemes, and constrains the latter to the domain of the former via the function uses. I.e., that $\forall n \in \mathcal{N}_S, \exists s \in \mathcal{S}$ such that $\mathsf{uses}(n, s)$.

The present ontology deals with three different types of scheme nodes, namely *rule of inference application nodes* (or *RA-nodes*), *preference application nodes* (or *PA-nodes*) and *conflict application nodes* (or *CA-nodes*). These are represented as three disjoint sets: $\mathcal{N}_S^{RA} \subseteq \mathcal{N}_S$, $\mathcal{N}_S^{PA} \subseteq \mathcal{N}_S$, and $\mathcal{N}_S^{CA} \subseteq \mathcal{N}_S$, respectively. The word ‘application’ on each of these types was introduced in the AIF as a reminder that these nodes function as instances, not classes, of possibly generic inference rules. Intuitively, $\mathcal{N}_S^{RA}$ captures nodes that represent (possibly non-deductive) rules of inference, $\mathcal{N}_S^{CA}$ captures applications of criteria (declarative specifications) defining conflict (e.g. among a proposition and its negation, among values, etc.), and $\mathcal{N}_S^{PA}$ are applications of (possibly abstract) criteria of preference among evaluated nodes.

	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 in supporting PA-node in conflict with another preference in PA-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

Table 1
Informal semantics of untyped edges in core AIF [11]

The AIF specification does not type its edges (which can increase processing cost). Instead, semantics for edges can be inferred when necessary from the types of nodes they connect. The informal semantics of edges are listed in Table 1. One of the restrictions imposed by the AIF is that no outgoing edge from an I-node can be directed directly to another I-node, i.e., $\nexists(i,j) \in \xrightarrow{edge}$ where both $i \in \mathcal{N}_I$ and $j \in \mathcal{N}_I$. This ensures that the type of any relationship between two pieces of information must be specified explicitly via an intermediate S-node. Bringing the above together, we present a formal definition of an argument network:

Definition 1 ***(Argument Network)***
An argument network Φ *is a graph consisting of:*

– *a set* $\mathcal{N}$ *of vertices (or* nodes*); and*
– *a binary relation* $\xrightarrow{edge}: \mathcal{N} \times \mathcal{N}$ *representing edges among nodes.*

such that $\nexists(i,j) \in \xrightarrow{edge}$ *where both* $i \in \mathcal{N}_I$ *and* $j \in \mathcal{N}_I$

A *simple* argument can be represented by linking a set of premises to a conclusion via a particular scheme. Formally:

Definition 2 ***(Simple Argument)***
A simple argument *in network* Φ *is a tuple* $\langle P, \tau, c\rangle$ *where:*

– $P \subseteq \mathcal{N}_I$ *is a set of nodes denoting premises;*
– $\tau \in \mathcal{N}_S^{RA}$ *is a node denoting a rule of inference application; and*
– $c \in \mathcal{N}_I$ *is a node denoting the conclusion;*

such that $\tau \xrightarrow{edge} c$, $\mathsf{uses}(\tau, s)$ *where* $s \in \mathcal{S}$, *and* $\forall p \in P$ *we have* $p \xrightarrow{edge} \tau$.

Following is an example description of a simple argument in propositional logic, depicted graphically in Figure 3(a). Note that to ease the reading of argument networks, we will distinguish S-nodes from I-nodes graphically by

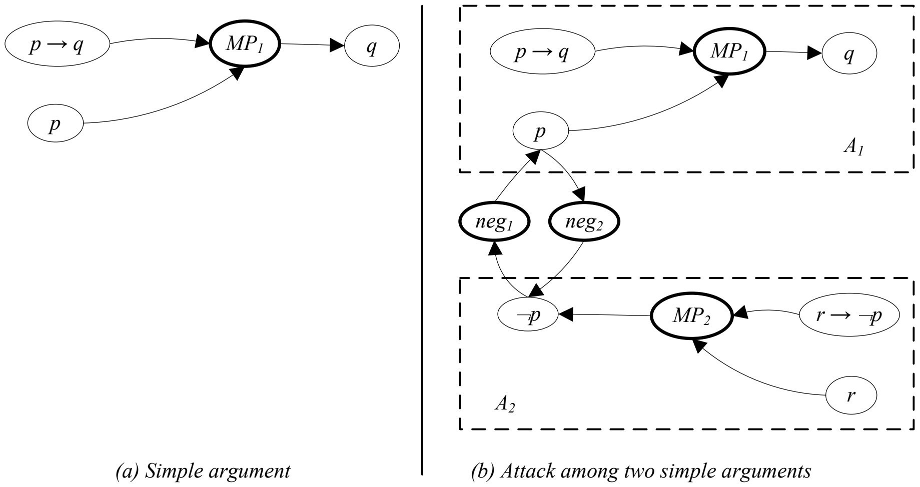


Fig. 3. Examples of simple arguments in an argument network

drawing the former with a slightly thicker border.

Example 1 ***(Simple Argument)***
The tuple $A_1 = \langle\{p, p \rightarrow q\}, MP_1, q\rangle$ is a simple argument in propositional language $\mathcal{L}$, where $p \in \mathcal{N}_I$ and $(p \rightarrow q) \in \mathcal{N}_I$ are nodes representing premises, and $q \in \mathcal{N}_I$ is a node representing the conclusion. In between them, the node $MP_1 \in \mathcal{N}_S^{RA}$ is a rule of inference application node (i.e., RA-node) that uses the modus ponens *natural deduction scheme, which can be formally written as follows:* $\mathsf{uses}(MP_1, \forall A, B \in \mathcal{L}\ \frac{A \quad A \rightarrow B}{B})$.

An attack or conflict from one information or scheme node to another information or scheme node is captured through a CA-node, which marks the type of conflict. The attacker is linked to the CA-node, and the CA-node is subsequently linked to the attacked node. Note that since edges are directed, each CA-node captures attack in one direction. Symmetric attack would require two CA-nodes, one in each direction. The following example describes a conflict, shown graphically in Figure 3(b), between two simple arguments.

Example 2 ***(Conflict among Simple Arguments)***
Recall the simple argument $A_1 = \langle\{p, p \rightarrow q\}, MP_1, q\rangle$. And consider another simple argument $A_2 = \langle\{r, r \rightarrow \neg p\}, MP_2, \neg p\rangle$. Argument A_2 undermines A_1 by supporting the negation of the latter's premise. This (symmetric) propositional conflict is captured through two CA-nodes labelled neg$_1$ and neg$_2$.

An important thing to note about the AIF is its ability to represent arguments at different levels of abstraction. For example, Dung's abstract argumentation framework [15] hides the internal structure of arguments, and only captures a single type of relation, which is a directed attack among whole arguments. This can be easily captured in the AIF. For example, the situation in Figure 3(b) can be captured by two nodes, labelled A_1 and A_2 and a CA-node in between directed edges from A_2 to A_1. It is also possible to define bridging rules connecting the different levels, allowing the system to, for example, infer the Dung relation from 3(b).

Note that S-to-S edges 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 kind of meta-justification for application of an inference rule or a particular criterion for defining preferences. Some instances of Toulmin backings [50], for example, could most accurately be captured through the use of RA-to-RA edges. If conflict between two I-nodes is captured via a CA-node, an RA-to-CA edge could encode some rationale of justifying the conflict specified in that CA-node (e.g., that each I-node linked by the CA-node specifies an alternative action for realising a goal; the CA-node expresses mutual exclusivity, and the justification, linked via the RA-node, corresponds to the reason that they cannot be carried out simultaneously).

4 Extending the Core AIF: Representing Argument Schemes

Argumentation schemes are *forms* of argument, representing stereotypical ways of drawing inferences from particular patterns of premises to conclusions. Schemes help categorise the way arguments are built. As such, they are referred to as *presumptive* inference patterns, in the sense that if the premises are true, then the conclusion may *presumably* be taken to be true.

Structures and taxonomies of schemes have been proposed by many theorists, such as Perelman and Olbrechts-Tyteca, [35], Grennan [21], Eemeren et al. [52], and Katzav and Reed [42]. But it is Walton's exposition [57] that has been most influential in computational work. Each *Walton scheme* type has a name, conclusion, set of premises and a set of critical questions bound to this scheme. Critical questions enable contenders to identify the weaknesses of an argument based on this scheme, and potentially attack the argument. A common example of Walton-style schemes is the 'Argument from Expert Opinion,' which takes the following form:

Example 3 ***(Scheme for Argument from Expert Opinion)***

- **Premise:** *Source E is an expert in the subject domain S.*
- **Premise:** *E asserts that proposition A in domain S is true.*
- **Conclusion:** *A may plausibly be taken to be true.*

Many other schemes were presented by Walton, such as *argument from consequence*, and *argument from analogy.* One can then identify *instances* that instantiate the scheme, such as the following example argument:

Example 4 ***(Instance of Argument from Expert Opinion)***

– **Premise:** *Allen is an expert in sport.*
– **Premise:** *Allen says that Brazil has the best football team.*
– **Conclusion:** *Presumably, Brazil has the best football team.*

With every scheme, Walton lays out a set of *critical questions*, which serve to inspect arguments based on this scheme more closely. For example, in the canonical scheme for '*Argument from expert opinion*,' there are six critical questions:

(1) *Expertise Question:* How credible is expert E as an expert source?
(2) *Field Question:* Is E an expert in the field that the assertion, A, is in?
(3) *Opinion Question:* Does E's testimony imply A?
(4) *Trustworthiness Question:* Is E reliable?
(5) *Consistency Question:* Is A consistent with the testimony of other experts?
(6) *Backup Evidence Question:* Is A supported by evidence?

As discussed by Prakken et al. [36] and Gordon and Walton [20], these questions are not all alike. The first, second, third and sixth questions refer to assumptions that the speaker makes, or, more accurately, *presumptions* required for the inference to go through (e.g., the critical question '*How credible is expert E as an expert source?*' questions a presumption by the proponent that '*Expert E is credible*'). The proponent of the argument retains the *burden of proof* if these questions are asked (e.g. the proponent must show evidence that expert E is credible). Numbers four and five, however, are somewhat different in that if asked, the burden of proof shifts, ceteris paribus, to the questioner (e.g., the opponent must demonstrate that another expert disagrees with E). These questions capture *exceptions* to the general rule, and correspond well to the *rebuttal* in Toulmin's [50] model of argument and its computational interpretation [44].

The Carneades model [20] is by far the most developed in terms of accounting representationally for these two distinct forms of implicit information present in schemes. We take a similar approach to Carneades in the sense that we distinguish explicitly between presumptions and exceptions. But our aim here is to offer an ontology of schemes and their component parts that builds on the AIF.

4.1 Defining Schemes in the AIF

Recall that in example 1, we represented the rule of inference application scheme in an RA-node labelled MP_1, and stated explicitly that it uses the

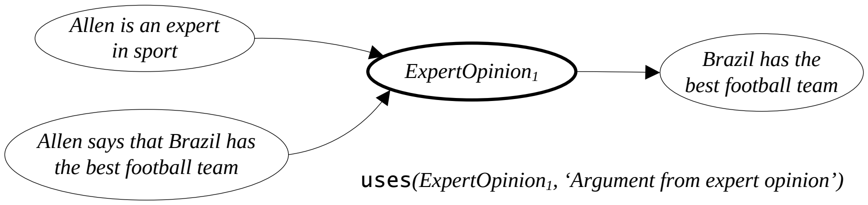

Allen is an expert in sport
Allen says that Brazil has the best football team
ExpertOpinion1
Brazil has the best football team
uses(ExpertOpinion1, 'Argument from expert opinion')

Fig. 4. An argument network showing an argument from expert opinion presented as a simple argument

modus ponens generic natural deduction rule. It would therefore seem natural to use the same approach with presumptive schemes. Attempting this approach with the argument from expert opinion from example 4 would lead to the argument described in Figure 4.

However, this approach is still somewhat limited, since it loses the information about the generic structure of the scheme. One way to deal with this is to supplement the RA-node with additional attributes that describe the various aspects of the scheme used: its conclusion type, premise types, critical questions, presumptions and exceptions. However, this would prohibit the re-use of these concepts in multiple arguments (since they would need to be copied for each instance of the scheme for argument from expert opinion). More significantly, this approach loses the explicit relationship between an actual premise and the generic *form* (or *descriptor*) it instantiates (e.g. that premise '*Allen is an expert in sport*' instantiates the generic form '*Source E is an expert in the subject domain S*'). To deal with this, we propose capturing the structure of the scheme explicitly in the argument network (i.e., we represent schemes themselves as inter-connected nodes). As we shall explain further below, this will prove useful in our implementation.

We will consider the set of schemes $\mathcal{S}$ as nodes in the argument network. Moreover, we introduce a new class of nodes, called *forms* (or *F-nodes*), captured in the set $\mathcal{N}_F \subseteq \mathcal{N}$, which is disjoint with the sets $\mathcal{N}_I$ and $\mathcal{N}_S$. Two distinct types of forms are presented: *premise descriptors* and *conclusion descriptors*. These are denoted by two disjoint sets: $\mathcal{N}_F^{Prem} \subseteq \mathcal{N}_F$ and $\mathcal{N}_F^{Conc} \subseteq \mathcal{N}_F$, respectively. Using these nodes, we can describe the structure of a presumptive inference scheme explicitly as part of the argument network itself. This is depicted in the shaded part of Figure 5.[7] With this in place, when we describe an actual presumptive argument, we can now explicitly link each node in the argument (the unshaded nodes) to the form node it instantiates (the shaded nodes), as can be seen in the example in Figure 5. Notice that here, we replaced the

[7] To improve readability, we will start using typed edges, which will enable us to explicitly distinguish between the different types of connections between nodes, as opposed to understanding the intended meaning of the edge implicitly based on the types of nodes it connects. All typed edges will take the form $\xrightarrow{type}$, where *type* is the type of edge, and $\xrightarrow{type} \subseteq \xrightarrow{edge}$.

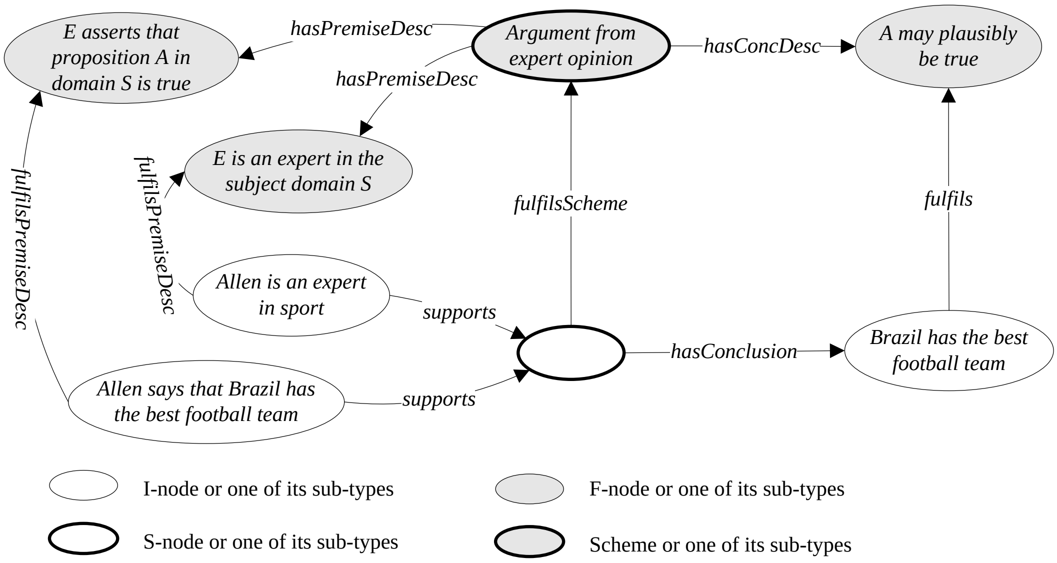

E asserts that proposition A in domain S is true
hasPremiseDesc
Argument from expert opinion
hasConcDesc
A may plausibly be true
hasPremiseDesc
E is an expert in the subject domain S
fulfilsPremiseDesc
fulfilsPremiseDesc
fulfilsScheme
fulfils
Allen is an expert in sport
supports
hasConclusion
Brazil has the best football team
Allen says that Brazil has the best football team
supports
I-node or one of its sub-types
F-node or one of its sub-types
S-node or one of its sub-types
Scheme or one of its sub-types

Fig. 5. An argument network showing an argument and a scheme description for the argument from expert opinion

predicate '**uses**' with the more specific edge $\xrightarrow{fulfilsScheme}: \mathcal{N}_S \times \mathcal{S}$.

The picture in Figure 5 is not yet complete, however, as it does not have any description of critical questions. Since each critical question corresponds either to a presumption or an exception, we only provide explicit descriptions (in the form of additional nodes) of the presumptions and exceptions associated with each scheme. With this in place, there is no longer any need to represent critical questions directly in the network, since they are inferable from the presumptions and exceptions, *viz.*, for every presumption or exception x, that scheme can be said to have a critical question '*Is it the case that x?*'

To express the scheme's typical presumptions, we add a new type of F-node called *presumption*, and represented by the set $\mathcal{N}_F^{Pres} \subseteq \mathcal{N}_F$. In the case of the argument from expert opinion, the three presumptions are shown at the lower part of Figure 6 and are all linked to the scheme via a new edge type $\xrightarrow{hasPresumption}: \mathcal{S} \times \mathcal{N}_F^{Pres}$.

As for representing exceptions, one alternative would be to view exceptions in exactly the same way and simply introduce a new type, as we have done for presumptions. The AIF, however, offers a much more powerful possibility. The clue comes from noting that exceptions function in a similar way to Toulmin's rebuttals: exceptions provide a way to challenge the use of an argument scheme. The function of challenging corresponds to the notion of a *conflict scheme* in the core AIF. In just the same way that stereotypical patterns of the passage of deductive, inductive and presumptive inference can be captured as rule of inference schemes, so too can the stereotypical ways of characterising conflict be captured as conflict schemes. Conflict, like inference,

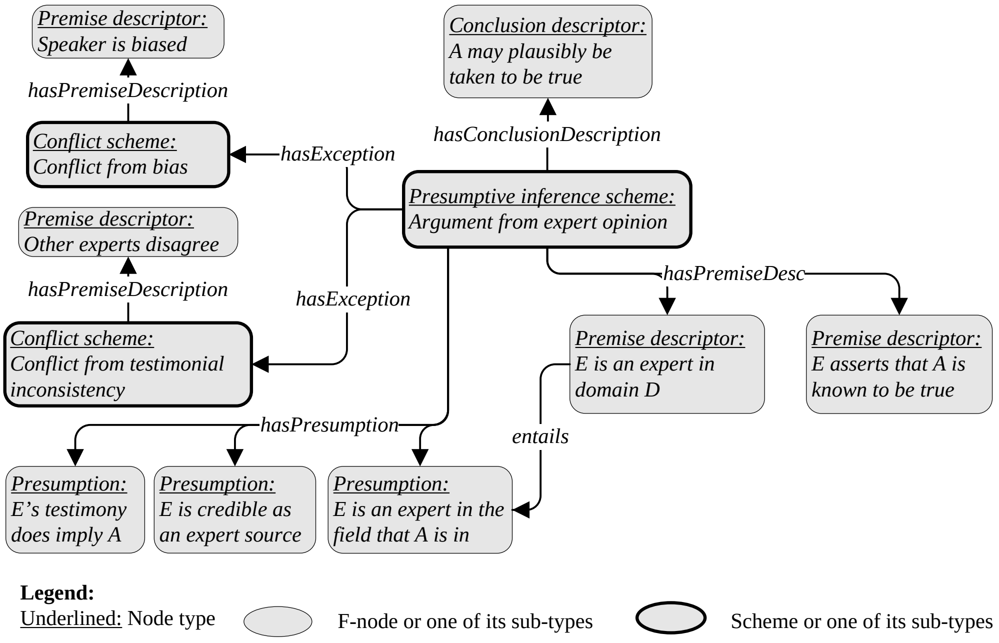

Premise descriptor: Speaker is biased
hasPremiseDescription
Conflict scheme: Conflict from bias
hasException
Conclusion descriptor: A may plausibly be taken to be true
hasConclusionDescription
Presumptive inference scheme: Argument from expert opinion
Premise descriptor: Other experts disagree
hasPremiseDescription
hasException
hasPremiseDesc
Conflict scheme: Conflict from testimonial inconsistency
Premise descriptor: E is an expert in domain D
Premise descriptor: E asserts that A is known to be true
entails
hasPresumption
Presumption: E's testimony does imply A
Presumption: E is credible as an expert source
Presumption: E is an expert in the field that A is in
Legend:
Underlined: Node type
F-node or one of its sub-types
Scheme or one of its sub-types

Fig. 6. An argument network showing the descriptions of the scheme for argument from expert opinion.

has some patterns that are reminiscent of deduction in their absolutism (such as the conflict between a proposition and its complement), as well as others that are reminiscent of non-deductive inference in their heuristic nature (such as the conflict between two courses of action with incompatible resource allocations). By providing a way to attack an argumentation scheme, exceptions can most accurately – and most expressively – be presented as conflict scheme descriptions. In the case of the argument from expert opinion, the three presumptions are are shown at the left part of Figure 6, all linked via a new edge type $\xrightarrow{hasException}$: $\mathcal{S} \times \mathcal{S}^C$. Note that each conflict scheme may have its own premise descriptors, or other forms describing its structure.

Finally, we note that in Walton's account of schemes, some presumptions are weakly related to certain premises. More specifically, a presumption may be implicitly or explicitly *entailed* by a premise. For example, the premise '*Source E is an expert in subject domain D*' entails the presumption that '*E is an expert in the field that A is in.*' While the truth of a premise may be questioned directly, questioning associated with the underlying presumptions can be more specific, capturing the nuances expressed in Walton's characterisation. We want to capture this relationship between some premises and presumptions explicitly, as it allows us to guide users in their critical questioning. Thus we have made use of a predicate ($\xrightarrow{entails}$: $\mathcal{N}_F^{Prem} \times \mathcal{N}_F^{Pres}$). Note, however, that not every presumption entails a particular premise, since some presumptions capture implicit assumptions underlying the whole scheme.

We can now formally provide a full definition of a presumptive inference scheme description.

Definition 3 ***(Presumptive Inference Scheme Description)***
A presumptive inference scheme description *in network* Φ *is a tuple* $\langle PD, \alpha, cd, \Psi, \Gamma, \xrightarrow{entails} \rangle$ *where:*

- $PD \subseteq \mathcal{N}_F^{Prem}$ *is a set of* premise descriptors*;*
- $\alpha \in \mathcal{S}^R$ *is the scheme;*
- $cd \in \mathcal{N}_F^{Conc}$ *is a* conclusion descriptor*.*
- $\Psi \subseteq \mathcal{N}_F^{Pres}$ *is a set of* presumption descriptors*;*
- $\Gamma \subseteq \mathcal{S}^C$ *is a set of* exceptions*; and*
- $\xrightarrow{entails}: \mathcal{N}_F^{Prem} \times \mathcal{N}_F^{Pres}$ *is a premise/presumption entailment relation;*

such that:

- $\alpha \xrightarrow{hasConcDesc} cd$*;*
- $\forall pd \in PD$ *we have* $\alpha \xrightarrow{hasPremiseDesc} pd$*;*
- $\forall \psi \in \Psi$ *we have* $\alpha \xrightarrow{hasPresumption} \psi$*;*
- $\forall \gamma \in \Gamma$ *we have* $\alpha \xrightarrow{hasException} \gamma$*;*

With the description of the scheme in place, we can now show how argument structures can be linked to scheme structures. In particular, we define a presumptive argument, which is an extension of the definition of a simple argument.

Definition 4 ***(Presumptive Argument)***
A presumptive argument *based on presumptive inference scheme description* $\langle PD, \alpha, cd, \Psi, \Gamma, \xrightarrow{entails} \rangle$ *is a tuple* $\langle P, \tau, c \rangle$ *where:*

- $P \subseteq \mathcal{N}_I$ *is a set of nodes denoting premises;*
- $\tau \in \mathcal{N}_S^{RA}$ *is a node denoting a rule of inference application; and*
- $c \in \mathcal{N}_I$ *is a node denoting the conclusion;*

such that:

- $\tau \xrightarrow{edge} c$*;* $\mathsf{uses}(\tau, \alpha)$*;*
- $\forall p \in P$ *we have* $p \xrightarrow{edge} \tau$*;*
- $\tau \xrightarrow{fulfilsScheme} \alpha$*;*
- $c \xrightarrow{fulfilsConclusionDesc} cd$*; and*
- $\xrightarrow{fulfilsPremiseDesc} \subseteq P \times PD$ *corresponds to a bijection (i.e. one-to-one correspondence) from* P *to* PD*.*

To show how these ontological structures govern and account for instantiated arguments, the next sub-section links the picture in Figure 6 to actual arguments generated by a simple dialogue.

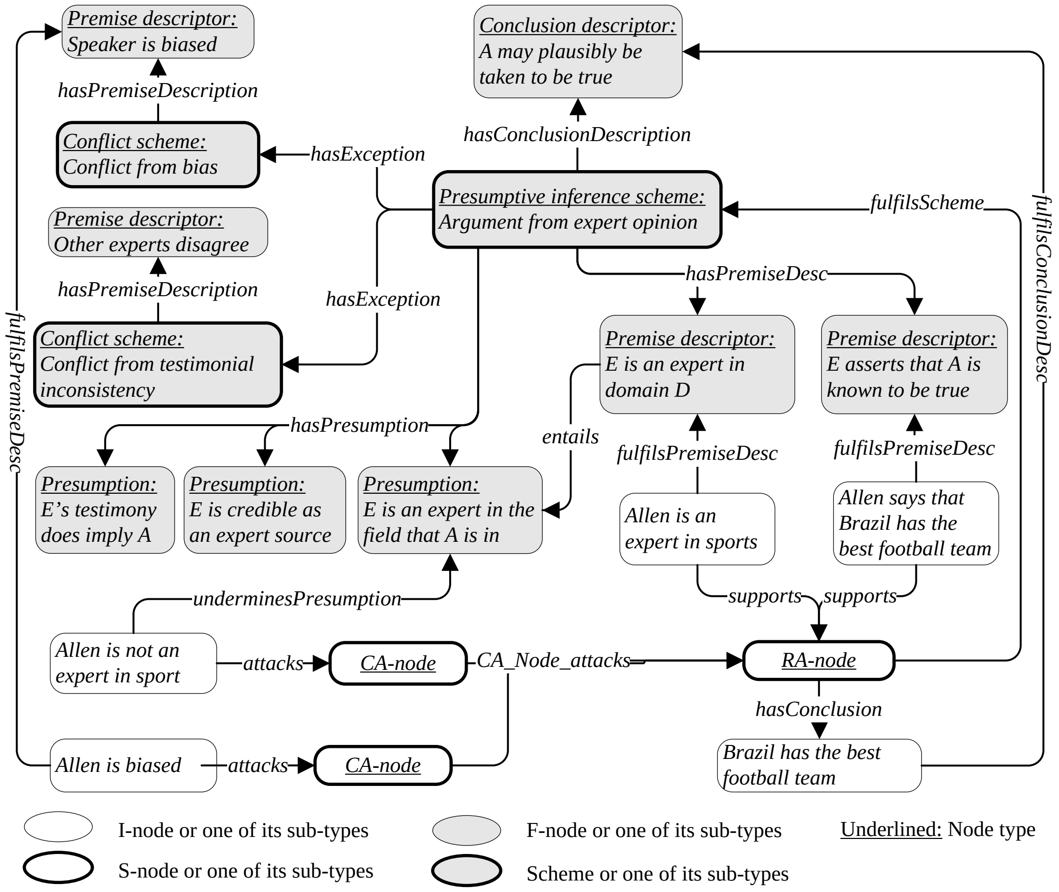


Fig. 7. An argument network showing an argument from expert opinion, two attackers arguments, and the descriptions of the schemes used by the argument and attackers. *Alice:* Brazil has the best football team: Allen is a sports expert and he says they do; *Bob:* Yes, but Allen is biased, and he is not an expert in sports!

4.2 An Example

Figure 7 shows arguments added to the scheme structure presented in Figure 6. It encodes the following arguments:

- **An argument from expert opinion:**
  - *Conclusion*: Brazil has the best football team
  - *Premise*: Allen says that Brazil has the best football team
  - *Premise*: Allen is an expert in sports
- **Two counter-arguments:**
  - *Undermine a presumption*: Allen is not an expert in sports;
  - *Point out an exception*: But Allen is biased

Figure 7 represents a surprisingly complex analysis for what appears to be a simple text. The reason for this is that the ontological superstructure needs to capture not only the content of the argument but also all the growth points at which new arguments might be added.

Note first that since presumptions correspond to hidden premises that are not stated explicitly in the argument [55], these presumptions are represented by scheme premise descriptors that are not fulfilled by any argument premise. The same goes with exceptions.

There are three distinct levels of analysis. At the bottom of Figure 7 (in unshaded boxes) are the components that instantiate real arguments – these are the actual premises, conclusions, inferences, conflicts and other components used in the expression of an argument. Further up in the figure (in shaded boxes) lies an intermediate level describing the types of inference (i.e. the scheme instance), the types of conflict (i.e. the conflict scheme instances) and the types of I-nodes (i.e. the presumptions, premise descriptors and conclusion descriptors). [8] Finally the ontological level is part of the AIF core and extended ontology, and is shown in Figure 8 below, which summarises our extensions to the original AIF ontologies presented earlier in Figure 2. [9] This layer simply views a presumptive inference scheme as a general class with many instances, presumption as a general class with many instances, and so on. The ontology level thus provides the types for nodes at the scheme description level, which in turn provides the specific analytical and generative material for the argument level. This tripartite approach is important to provide an AIF ontology that is both implementable in the form of software tools for argument construction and analysis, and also able to interact with other AIF extensions that make use of different description level data (e.g., different scheme sets).

5 AIF-RDF: The Extended AIF Ontology in RDF Schema

In this section, we describe AIF-RDF: an implementation of the core AIF and our extensions using the RDF Schema computational ontology language.

5.1 RDF & RDFS

The Resource Description Framework (RDF) [26] is a meta-data model based on the idea of making *statements* about *resources*. A resource has a unique Universal Resource Identifier (URI), and can be considered as a physical entity (e.g. an electronic document like a picture or a file), or a concept (e.g. a

[8] This level would also include PreferenceScheme instances if there were any.

[9] To simplify the figure, we ommitted some details that are irrelevant to our extension, such as the context.

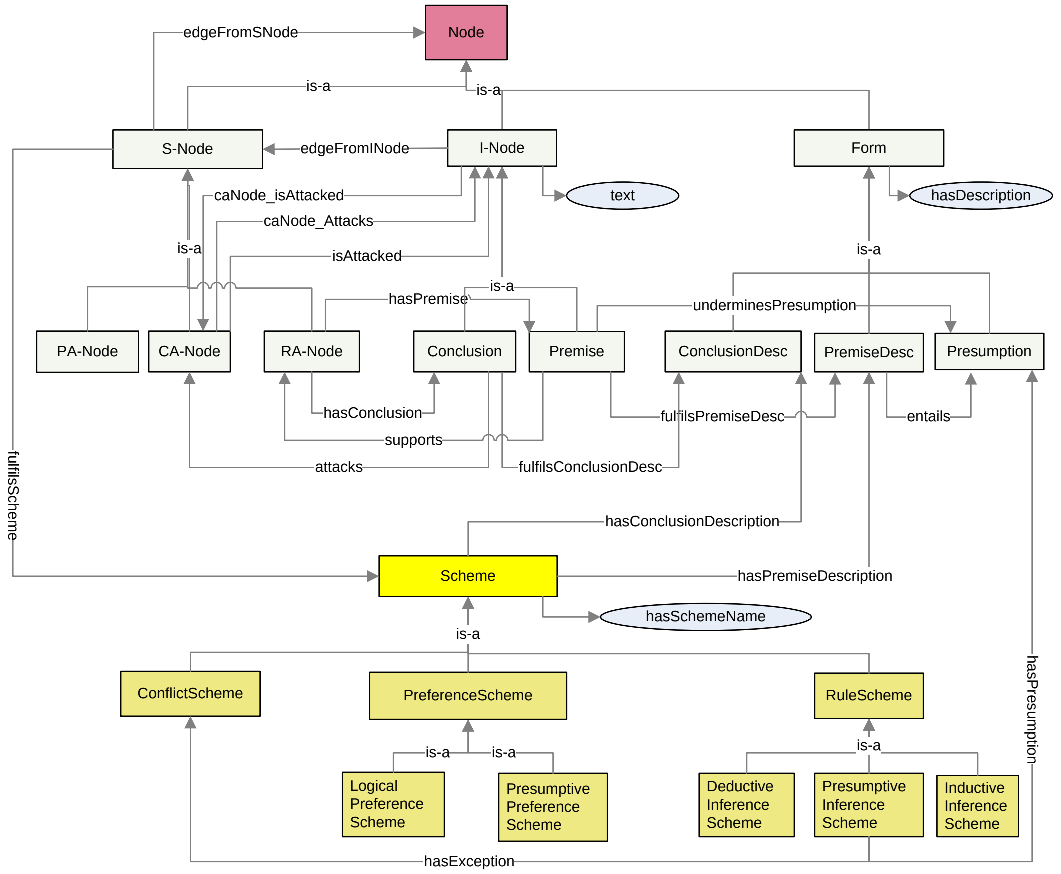
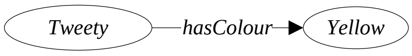


Fig. 8. Extensions to the original AIF

person or a medical term). A statement is a *subject-predicate-object* expression, sometimes called a *triple*. The subject denotes the resource that is being described by the statement. The predicate describes the relationship between the subject and the object. The object can be another resource (with its own URI) or a *literal* (e.g. a string of text). RDF statements can be captured in different syntactic formats. For example, the statement '*Tweety has a yellow colour*' can be written as an RDF triple:

```
(Tweety, hasColour, Yellow)
```

or as a directed labelled graph:



or in the following RDF/XML format:

```
<rdf:Description rdf:about=Tweety>
  <rdf:hasColour>
    <rdf:Description rdf:about=Yellow/>
  </rdf:hasColour>
</rdf:Description>
```

RDF Schema (RDFS) [5] is an XML based knowledge representation language, built on top of RDF, that allows the definition of domain ontologies (concepts and the relationships between them). It provides a specific set of constructs for specifying classes and class hierarchies (or taxonomies), properties (or predicates) and property hierarchies, restrictions on the domains and ranges of properties, and so on. RDFS specifications are themselves RDF statements. For example, the triple `(Person, rdfs:subClassOf, Agent)` specifies that the class 'Person' is a sub-class of the class 'Agent.' Finally, it is possible to make statements that link domain resources to domain ontological specification by combining RDF and RDFS. For example, the following RDF/XML code states that resource 'Tweety' is an instance of class 'Bird:'

```
<rdf:Description rdf:about=Tweety>
  <rdf:type rdf:resource=Bird/>
</rdf:Description>
```

Below, we describe the implementation of our extended AIF ontology in RDFS, which enables us to specify argument networks in the same way as RDF graphs are described. When compared with pure XML, there are a number of important features of RDF and RDFS that are relevant to our aims:

- When compared with XML, RDFS provides a more concise and standard way of describing extensible domain ontologies, which is convenient for describing an ontology like the AIF and extensions thereof;
- RDF's model is based on describing statements about resources in the form of directed graphs, while XML is based on describing (tree-like) document structures. A graphical model is more suitable for representing (and potentially visualising) argument networks;
- Querying an XML tree that represents relational knowledge can be very complex because there are, in general, many ways in which a logical specification can be described in XML, and the query written has to be independent of the syntactic choice made. RDF provides standard ways of writing statements so that however they occur in a document, they produce the same effect in RDF terms. So querying RDF statements can be done more easily through a query language (e.g. RQL) and associated engine that *understands* the RDF data model and can retrieve the correct results regardless of the (XML-based or other) syntactic variant in which RDF statements are written [1, Chapter 3];
- The graph concept and the subject-object relationship in RDF makes manipulating network structures (e.g. argument networks) easy. This is done through the insertion and deletion of triples, without having to worry about the order of the statements inserted, or the variety of syntactic variants for representing those statements.

In the following subsection, we show how RDFS and RDF can be used to capture our ontology and its argument instances.

5.2 The Extended AIF in RDFS/RDF

In this section, we briefly describe the implementation of our extended AIF ontology in RDFS code. The implementation was done using Protégé [34], an ontology building tool that supports Semantic Web languages such as RDFS.

The extended AIF ontology described in Figure 8 was implemented as follows. The various node types are represented as a hierarchy of classes, and edges connecting nodes are represented as class attributes. For example, the following RDFS code defines the class *I-Node* and states that it is a subclass of *node*.

```
<rdf:Description rdf:about="http://protege.stanford.edu/kb#I-Node">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdfs:label>I-Node</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://protege.stanford.edu/kb#Node"/>
</rdf:Description>
```

In our implementation, all edges are explicitly typed, in order to make querying easier. The constraints on edges specified by the AIF are represented using the domain and range attributes. Below is an RDFS representation of edges emanating from S-nodes:

```
<rdf:=Description rdf:about="edgeFromSNode">
    <rdf:type rdf:resource="Property"/>
    <a:minCardinality> 1 </a:minCardinality>
    <rdfs:label> edgeFromSNode </rdfs:label>
    <rdfs:range rdf:resource="Node"/>
    <rdfs:domain rdf:resource="S-Node"/>
    <rdfs:subPropertyOf rdf:resource="edge"/>
</rdf:Description>
```

Recall that the core AIF requires that all classes of edges and nodes are disjoint (e.g. a node cannot be of type I-Node and S-Node at the same time). Unfortunately, disjointedness cannot be expressed in RDFS, and considered one of the limitations of this semantic language.

Details of the fully encoded AIF-RDF can be found on ArgDF's Web site (`http://www.argdf.org/source/`).

6 ArgDF: A System for Authoring and Navigating Arguments

ArgDF is a pilot Semantic Web-based system that uses the AIF-RDF ontology presented in the previous section. ArgDF enables users to create and query arguments that are semantically annotated using different argumentation schemes. The system also allows users to manipulate arguments by attacking or supporting parts of existing arguments, and also to re-use exist-

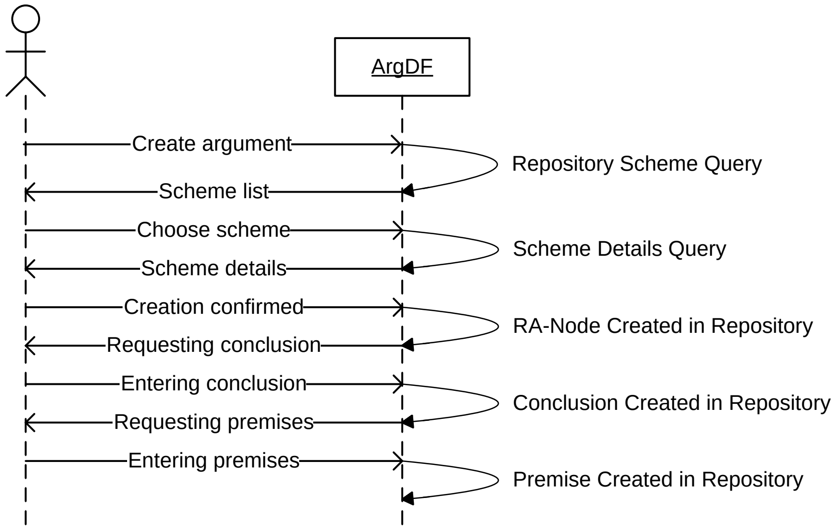


Fig. 9. New Argument Creation Cycle

ing parts of an argument in the creation of new arguments. ArgDF also allows users to create new argumentation schemes. As such, ArgDF is an open platform not only for representing arguments, but also for building interlinked and dynamic argument networks. In the remainder of this Section, we describe the system in detail.

It is worth noting that the system only acts as a demonstrator of the basic functionality enabled by our framework. We envisage a variety of more feature-rich systems that may be built using the same framework, as we shall discuss in Section 7.

6.1 ArgDF Platform Overview

ArgDF uses a variety of software components such as the Sesame RDF repository [6],[10] PHP scripting, XSLT, the Apache Tomcat server,[11] and MySQL database. The system also uses Phesame,[12] a PHP class containing a set of functions for communicating with Sesame through PHP pages. The Sesame RDF repository offers the central features needed by the system, namely: (i) uploading RDF and RDFS single statements or complete files; (ii) deleting RDF statements; (iii) querying the repository using standard Semantic Web query languages; and (iv) returning RDF query results in a variety of computer processable formats including XML, HTML or RDF. Sesame is well-supported and has been used in a variety of Semantic Web-based systems.

[10] See also: `http://www.openrdf.org`

[11] `http://tomcat.apache.org/`

[12] `http://www.hjournal.org/phesame`

6.2 Creating New Arguments

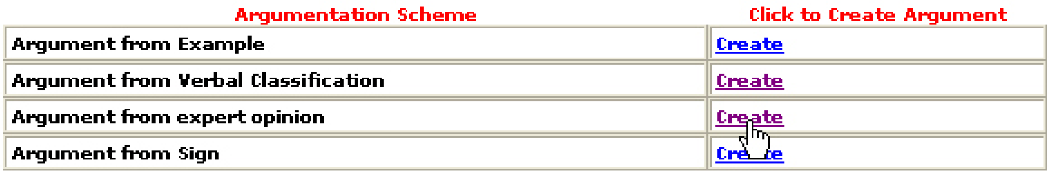
In ArgDF, a user can create new arguments based on existing argument schemes. The system lists the available argument schemes, and allows the user to choose the scheme to which the argument belongs. Details of the argumentation scheme selected are then retrieved from the repository, and the generic form of the argument is displayed to the user to guide the creation of the conclusion and premises.

In the background, the creation of a new argument involves many processes, ranging from the upload of RDF statements, to querying the repository and displaying information to the end user. Figure 9 visualises the steps to give a clearer idea of the complete cycle in a UML sequence diagram. We explain the process in more detail below.

Whenever there is a screen in ArgDF in which there is a list of options for the user to choose from, there will be two queries that will be applied to the repository: one to extract the text and details of the resources, and another query to extract the labels and URIs. These queries are written using the RDF Query Language (RQL) [24], which is supported by Sesame. RQL queries are similar to database queries and take the form *Select-From-Where*. For example, querying the ArgDF repository to extract the name of the schemes can be done through the following RQL query:

```
select Scheme, PresumptiveInferenceScheme-hasSchemeName
from   Scheme : kb:PresumptiveInferenceScheme kb:hasSchemeName
       PresumptiveInferenceScheme-hasSchemeName
using namespace
       rdf = http://www.w3.org/1999/02/22-rdf-syntax-ns# ,
       rdfs = http://www.w3.org/2000/01/rdf-schema# ,
       kb = http://protege.stanford.edu/kb#
```

This query is passed to the Sesame server using Phesame and the returned result, in XML format, is then rendered as HTML by two XSLT transforms. The first XSLT manipulates the hyperlink to enable subsequent argument navigation tasks by the user. The second XSLT displays the name of the schemes in a table. For example, the result of the RQL query above can be passed in XSLT to produce the HTML output shown in Figure 10.



Argumentation Scheme	Click to Create Argument
Argument from Example	Create
Argument from Verbal Classification	Create
Argument from expert opinion	Create
Argument from Sign	Create

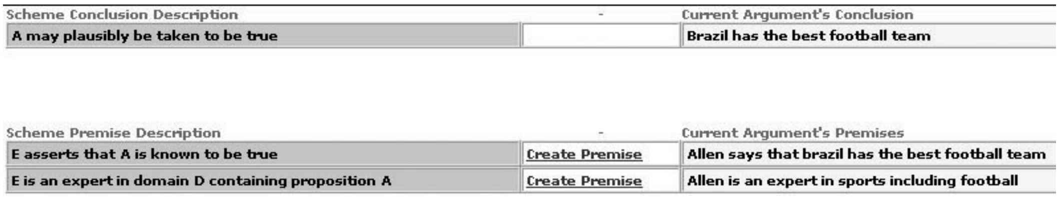
Fig. 10. XSLT Table Output

After choosing the scheme, the Uniform Resource Identifier (URI) of the instance scheme is passed to the next page, and then again two queries are performed: one extracts the conclusion's text of the scheme instance matching the URI of the one chosen by the user, and the other extracts its premises' text. The scheme details are then rendered using two XSLT files applied during all the argument creation process.

The first element ArgDF will upload to the repository is the RA-node: the scheme node that will hold the various argument pieces together. This process happens automatically before creating the conclusion and the premises. A unique URI is applied to the RA-node instance, and is linked to the URI of the previously chosen scheme using the *fulfilsScheme* relationship. This links the newly created argument to the scheme chosen by the user. The RDF code uploaded to Sesame for the creation of the RA-Node looks like this:

```
<rdf:RDF>
  <kb:RA-Node rdf:about=&kb;MySQL URI Generation
              rdfs:label=MySQL URI Generation>
    <kb:fulfilsScheme rdf:resource=&kb;Selected Scheme/>
  </kb:RA-Node>
</rdf:RDF>
```

After uploading the RA-Node RDF statement, the user will be redirected to enter the conclusion and the premises of the argument. The system guides the user during this process based on the scheme structure (selected earlier by the user). The conclusion and premises instances will get a unique URI, and will be linked to the previously created RA-Node. In addition, each argument conclusion and premise entered by the user must fulfil the conclusion and premise description of the scheme as shown in Figure 11. Thus, both the argument structure and scheme structure are generated in the background and encoded in RDF.



Scheme Conclusion Description	-	Current Argument's Conclusion
A may plausibly be taken to be true		Brazil has the best football team

Scheme Premise Description	-	Current Argument's Premises
E asserts that A is known to be true	Create Premise	Allen says that brazil has the best football team
E is an expert in domain D containing proposition A	Create Premise	Allen is an expert in sports including football

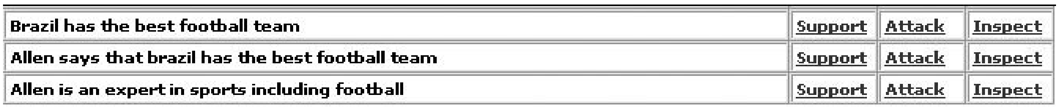
Fig. 11. Argument Creation in ArgDF

6.3 Support, Attack and Search of Existing Arguments

ArgDF allows users to support and/or attack existing expressions. The list of existing expressions in the repository can be displayed as shown in Figure 12. The user can choose the statement they want to support or attack. Both conclusions and premises can be supported and attacked in this way. When a user chooses to support an existing premise, this premise will have two roles:

as a premise in one argument, and as a conclusion in another one. Thus, the system allows for the *chaining* of arguments.

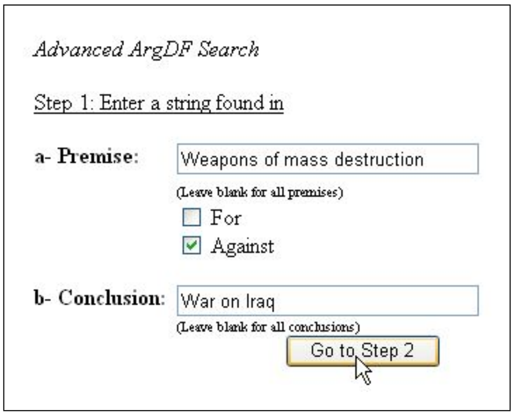
To support existing expressions, the user can create supporting premises after choosing a scheme to be used in the support. Similarly, to attack, the user selects a conflict scheme and introduces a new expression that fulfils the conflict. That new expression can then be the conclusion of a new argument, and so on.



Brazil has the best football team	Support	Attack	Inspect
Allen says that brazil has the best football team	Support	Attack	Inspect
Allen is an expert in sports including football	Support	Attack	Inspect

Fig. 12. Listing Existing Claims

The system also enables users to search existing arguments, by specifying text found in the premises or the conclusion, as well as the type of relationship between these two (i.e. whether it is a support or an attack). For example, Figure 13 shows the first step of the interface with a query asking for arguments against the war on Iraq, and which mention ‘*weapons of mass destruction*’ in their premises. The following step (not shown here) would then ask the user to filter arguments based on the scheme used. For example, the user can specify that they are only interested in arguments based on expert opinion. In the background, the system uses this information to construct an RQL query which is then submitted to the RDF repository.



Advanced ArgDF Search

Step 1: Enter a string found in

a- Premise: Weapons of mass destruction
(Leave blank for all premises)
□ For
☑ Against

b- Conclusion: War on Iraq
(Leave blank for all conclusions)

Fig. 13. Argument search interface

6.4 Linking Existing Premises to a New Argument

While creating premises supporting a given conclusion through a new argument, the user can *re-use* existing premises from the system. This functionality can be useful, for example, in Web-based applications that allow users to use existing Web content (e.g. a news article, a legal document) to support new or existing claims. This way a premise can be used for two or more different arguments. The resulting network structure is exemplified in Figure 14, in which a single claim constitutes a premise for two arguments, in a *divergent* argumentation structure.

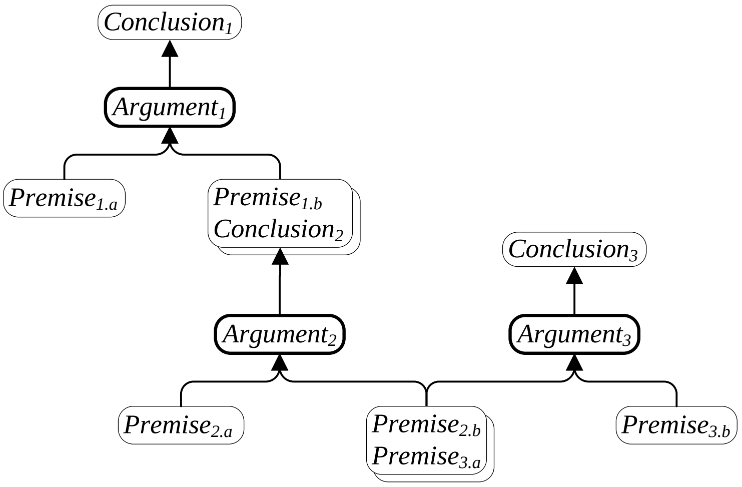


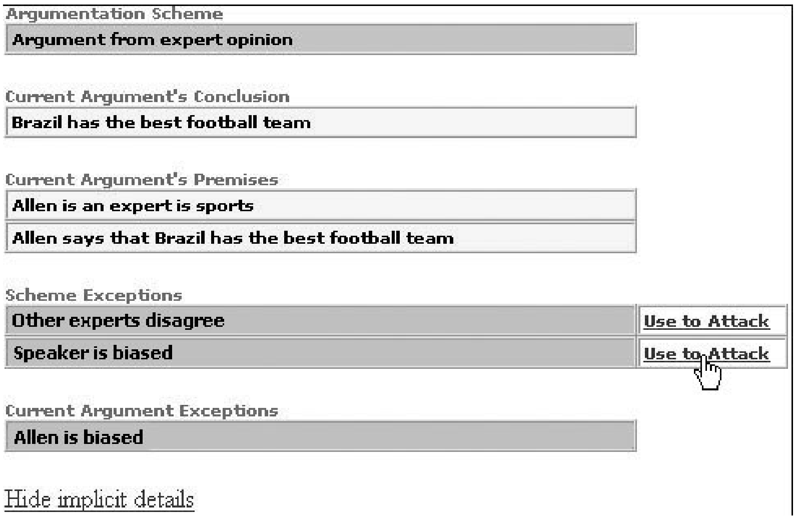
Fig. 14. Chaining of arguments 1 and 2, and shared premise in arguments 2 and 3

6.5 Attacking Arguments through Implicit Assumptions

With our account of presumptions, premises and exceptions, it becomes possible to construct an automatic mechanism for *presuming*. Consider a case in which a user constructs an argument using a scheme which has presumptions, but fails to explicitly add premises corresponding to those presumptions. It could be that this scenario is quite common –after all, presumptions are usually presumed, by definition, rather than stated. In this case, it is a simple matter to identify the fact that there are presumptions in the scheme which do not correspond to explicit premises.

With the system explicitly performing the act of 'presuming' in this way, the argument can be presented to the user with the presumptions made accessible, allowing for challenge or exploration of those presumptions by which the argument inference is warranted. A similar approach can be taken to *exceptions* to the application of a scheme. The system can make these explicit, allowing for attacks on existing arguments. This is exactly the role that Walton envisaged for his critical questions [57]. And ArgDF exploits knowledge about such implicit assumptions (namely presumptions and exceptions) in order to enable richer interaction between the user and the arguments.

ArgDF allows the user to *inspect* existing claims by displaying all the arguments in which this claim is involved: being a conclusion or a premise supporting a conclusion. After opening an argument, exceptions and presumptions can be opened leading the way for an implicit attack of the argument either through an exception (as in Figure 15), or through undermining a presumption (as in Figure 16).



Argumentation Scheme

Argument from expert opinion

Current Argument's Conclusion

Brazil has the best football team

Current Argument's Premises

Allen is an expert is sports

Allen says that Brazil has the best football team

Scheme Exceptions

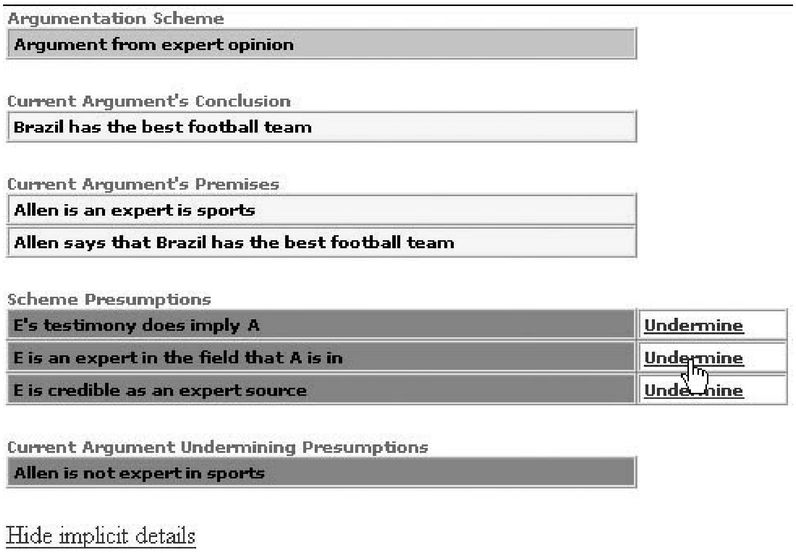
Other experts disagree	Use to Attack
Speaker is biased	Use to Attack

Current Argument Exceptions

Allen is biased

Hide implicit details

Fig. 15. Implicit Attack Through an Exception in ArgDF



Argumentation Scheme

Argument from expert opinion

Current Argument's Conclusion

Brazil has the best football team

Current Argument's Premises

Allen is an expert is sports

Allen says that Brazil has the best football team

Scheme Presumptions

E's testimony does imply A	Undermine
E is an expert in the field that A is in	Undermine
E is credible as an expert source	Undermine

Current Argument Undermining Presumptions

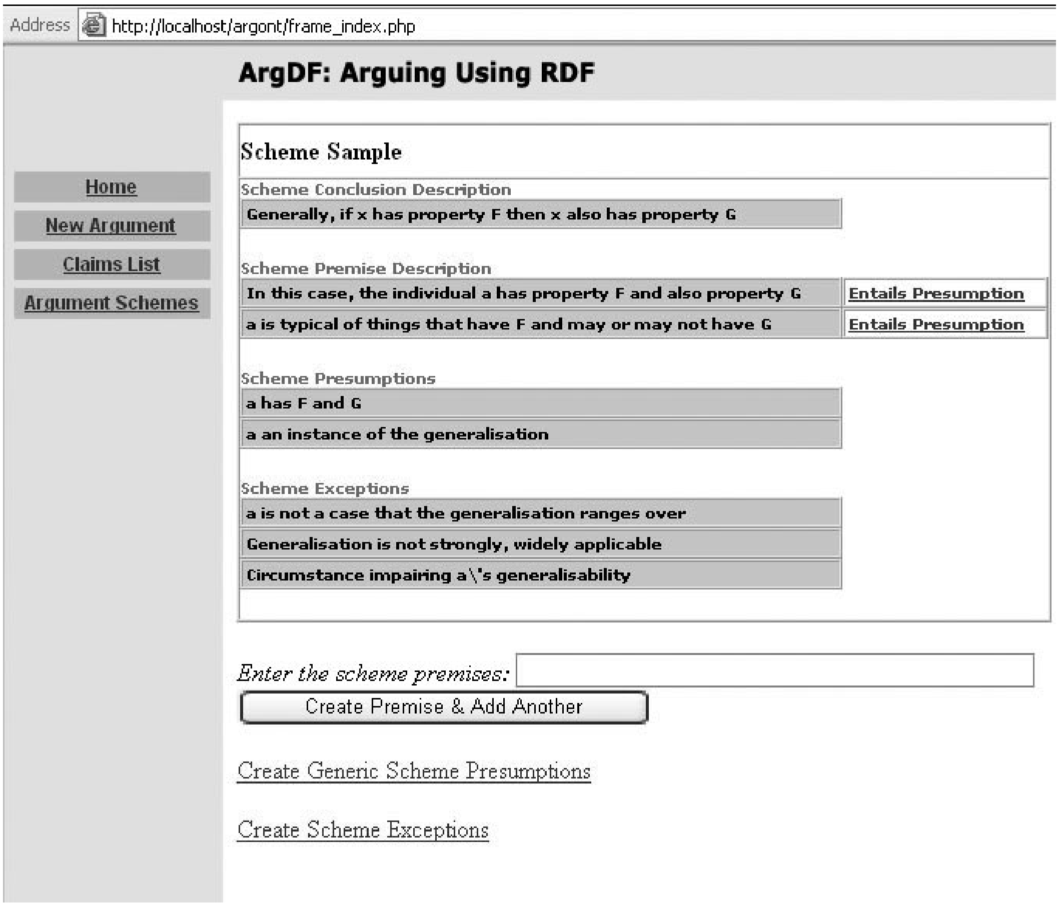
Allen is not expert in sports

Hide implicit details

Fig. 16. Implicit Attack Through Undermining a Presumption in ArgDF

6.6 Creation of New Schemes

The user can also create new argumentation schemes through the interface of ArgDF without having to modify the ontology itself, because actual schemes are simply instances of the 'Scheme' class. Figure 17 shows a screen shot of the creation 'Argument from Example' scheme in ArgDF.


Address http://localhost/argont/frame_index.php
Home
New Argument
Claims List
Argument Schemes
Create Premise & Add Another
Create Generic Scheme Presumptions
Create Scheme Exceptions

ArgDF: Arguing Using RDF

Scheme Sample

Scheme Conclusion Description

Generally, if x has property F then x also has property G

Scheme Premise Description

In this case, the individual a has property F and also property G	Entails Presumption
a is typical of things that have F and may or may not have G	Entails Presumption

Scheme Presumptions

a has F and G

a an instance of the generalisation

Scheme Exceptions

a is not a case that the generalisation ranges over

Generalisation is not strongly, widely applicable

Circumstance impairing a\'s generalisability

Enter the scheme premises:

Fig. 17. Creating a new Scheme for 'Argument from Example' in ArgDF

7 Conclusions and Future Possibilities

As tools for electronic argumentation grow in sophistication, number and popularity, so the role for the AIF and its implementations are expected to become more important. What this paper has done is to sketch where this trend takes us – the World Wide Argument Web – and to describe some of the technical components that will support it, building on a foundation of Walton's theory, the AIF ontology and the Semantic Web.

In Section 2.3, we introduced desiderata necessary for the creation of a WWAW. We now revisit them and reflect on how our framework, its specification in the AIF-RDF ontology, and its realisation in the ArgDF system, all measure up to those desiderata.

(1) *The WWAW must support the storage, creation, update and querying of argumentative structures:* ArgDF is a Web-based system that supports the storage, creation, update and querying of argument data structures based on Walton's argument schemes. Though the prototype implementation employs a centralised server, the model can support large-scale

distribution.

(2) *The WWAW must have Web-accessible repositories:* Arguments are uploaded on a Sesame RDF repository which can be accessed and queried openly through the Web and using a variety of RDF standard query languages.

(3) *The WWAW language must be based on open standards, enabling collaborative development of new tools:* Arguments in the ArgDF system are annotated in RDF using ontologies defined using the RDF Schema ontology language, both of which are open standards endorsed by the W3C. A variety of software development tools can be used for taking advantage of this.

(4) *The WWAW must employ a unified, extensible argumentation ontology:* Our ontology captures the main concepts in the Argument Interchange Format ontology [11], which is the most current general ontology for describing arguments and argument networks.

(5) *The WWAW must support the representation, annotation and creation of arguments using a variety of argumentation schemes:* AIF-RDF preserves the AIF's strong emphasis on scheme-based reasoning patterns, conflict patterns and preference patterns, and is designed specifically to accommodate extended and modified scheme sets.

Together, the AIF-RDF ontology implementation and the ArgDF software tool demonstrate how the WWAW can be put together. AIF represents a first step towards an open, flexible and re-usable mechanism for handling argumentation in a wide variety of domains, but the high level of abstraction that was demanded of it also presents challenges to developers' abilities to use it. AIF-RDF tackles those challenges and bridges the gap between the ontological abstraction and the code-level detail. ArgDF then demonstrates the flexibility that AIF-RDF affords, and in particular, offers an example of rapid tool development on the basis of theoretical advances in the understanding of argument structure: the result is a functionally intuitive argumentational interface to slippery concepts such as exceptions and presumptions. In this way, ArgDF represents an exemplar for developers as the WWAW starts to grow and provide real services for the online community. Following are some potential usage scenarios that may exploit the infrastructure presented in this paper.

Question Answering: An obvious extension of the current system is to exploit the variety of ideas and techniques for improving question answering by exploiting features of the Semantic Web [32]. Prospects range from using query refinement techniques to interactively assist users find arguments of interest through Web-based forms, to processing natural language questions like '*List all arguments that support the War on Iraq on the basis of expert assessment that Iraq has Weapons of Mass Destruction (WMDs).*' This functionality would be more significant if AIF-RDF became more widely used, resulting in annotations of a large amount content on the Web. Translating the ontology to more expressive Semantic Web ontology languages such as OWL [33] can also enable ontological reasoning over argument structures, for example, to automatically classify arguments, or to identify semantic similarities among arguments.

Interface and argument visualisation: ArgDF itself provides only rudimentary graphical displays. The visual sophistication of systems like Reason!Able [17], ClaiMaker [7], and Araucaria [45] will represent a bare minimum if the WWAW is to appeal to non-experts. Contributing new arguments must be as simple and intuitive as blogging is; connecting to other people's arguments must be as easy as social bookmarking is.

Argumentative Blogging: Another potential extension is combining our framework with so-called *Semantic Blogging* tools [9], to enable users to annotate their blog entries as argument structures for others to search, and to blog in response to one another's arguments. This can represent a useful approach for building up large amounts of annotations, which would in turn make the question answering scenario mentioned above more viable.

Mass-collaborative argument editing: Another approach to accumulating argument annotations is through mass-collaborative editing of semantically connected argumentative documents in the style of *Semantic Wikipedia* [56]. A basic feature of this kind is already offered by Discourse DB (discussed above in Section 2), which has started accumulating sizable content.

All these future directions represent extensions to the basic, core idea. What has been presented here is a clearly specified, and (at least in prototype form) implemented foundation upon which the WWAW can be brought into existence, piece by piece.

Appendix: Sample Argument in AIF-RDF

The below code, extracted from the Sesame RDF server, represents 2 arguments under attack created in ArgDF. The purpose of this appendix is to show in full how the resources are inter-connected in RDF. Resources have unique identifications, with a certain type like 'premise' and specific attributes which can either be literals such as 'text,' or relationships heading to other resources such as the 'supports' relationship.

The code flows by representing the first argument's premises, conclusion and RA-Node. Then the CA-Node, linking the arguments in conflict is presented, followed by the second argument's RA-Node, attacking the former one, as well as its premises and conclusion.

```
<rdf:Description rdf:about="http://protege.stanford.edu/kb#ArgOnt_Instance_16">
  <rdf:type rdf:resource="http://protege.stanford.edu/kb#Premise"/>
  <kb:text>Allen says that Brazil has the best football team</kb:text>
  <rdfs:label>ArgOnt_Instance_16</rdfs:label>
  <kb:supports rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_13"/>
  <kb:edgeFromINode rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_13"/>
  <kb:fulfilsPremiseDesc rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_6"/>
</rdf:Description>

<rdf:Description rdf:about="http://protege.stanford.edu/kb#ArgOnt_Instance_15">
  <rdf:type rdf:resource="http://protege.stanford.edu/kb#Premise"/>
  <kb:text>Allen is an expert is sports</kb:text>
  <rdfs:label>ArgOnt_Instance_15</rdfs:label>
  <kb:supports rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_13"/>
  <kb:edgeFromINode rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_13"/>
  <kb:fulfilsPremiseDesc rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_7"/>
</rdf:Description>

<rdf:Description rdf:about="http://protege.stanford.edu/kb#ArgOnt_Instance_14">
  <rdf:type rdf:resource="http://protege.stanford.edu/kb#Conclusion"/>
  <kb:text>Brazil has the best football team</kb:text>
  <rdfs:label>ArgOnt_Instance_14</rdfs:label>
  <kb:edgeFromINode rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_13"/>
  <kb:fulfilsConclusionDesc rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_5"/>
  <kb:CANode_isAttacked rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_50486"/>
  <kb:edgeFromINode rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_50486"/>
</rdf:Description>

<rdf:Description rdf:about="http://protege.stanford.edu/kb#ArgOnt_Instance_13">
  <rdf:type rdf:resource="http://protege.stanford.edu/kb#RA-Node"/>
  <rdfs:label>ArgOnt_Instance_13</rdfs:label>
  <kb:hasConclusion rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_14"/>
  <kb:edgeFromSNode rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_14"/>
  <kb:hasPremise rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_15"/>
  <kb:edgeFromSNode rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_15"/>
  <kb:hasPremise rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_16"/>
  <kb:edgeFromSNode rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_16"/>
  <kb:fulfilsScheme rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_4"/>
</rdf:Description>

<rdf:Description rdf:about="http://protege.stanford.edu/kb#ArgOnt_Instance_50486">
  <rdf:type rdf:resource="http://protege.stanford.edu/kb#CA-Node"/>
  <rdfs:label>ArgOnt_Instance_50486</rdfs:label>
  <kb:CANode_Attacks rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_14"/>
  <kb:edgeFromSNode rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_14"/>
  <kb:isAttacked rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_50487"/>
</rdf:Description>

<rdf:Description rdf:about="http://protege.stanford.edu/kb#ArgOnt_Instance_50485">
  <rdf:type rdf:resource="http://protege.stanford.edu/kb#RA-Node"/>
  <rdfs:label>ArgOnt_Instance_50485</rdfs:label>
  <kb:fulfilsScheme rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_4"/>
  <kb:hasConclusion rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_50487"/>
  <kb:edgeFromSNode rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_50487"/>
  <kb:hasPremise rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_50488"/>
  <kb:edgeFromSNode rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_50488"/>
  <kb:hasPremise rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_50489"/>
  <kb:edgeFromSNode rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_50489"/>
</rdf:Description>

<rdf:Description rdf:about="http://protege.stanford.edu/kb#ArgOnt_Instance_50487">
  <rdf:type rdf:resource="http://protege.stanford.edu/kb#Conclusion"/>
  <kb:text>Germany has the best football team</kb:text>
  <rdfs:label>ArgOnt_Instance_50487</rdfs:label>
  <kb:edgeFromINode rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_50485"/>
  <kb:attacks rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_50486"/>
  <kb:edgeFromINode rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_50486"/>
  <kb:fulfilsConclusionDesc rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_5"/>
</rdf:Description>

<rdf:Description rdf:about="http://protege.stanford.edu/kb#ArgOnt_Instance_50489">
  <rdf:type rdf:resource="http://protege.stanford.edu/kb#Premise"/>
  <kb:text>Jim is an expert in sports including football</kb:text>
  <rdfs:label>ArgOnt_Instance_50489</rdfs:label>
  <kb:edgeFromINode rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_50485"/>
  <kb:supports rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_50485"/>
  <kb:fulfilsPremiseDesc rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_7"/>
</rdf:Description>

<rdf:Description rdf:about="http://protege.stanford.edu/kb#ArgOnt_Instance_50488">
  <rdf:type rdf:resource="http://protege.stanford.edu/kb#Premise"/>
  <kb:text>Jim says that Germany has the best football team</kb:text>
  <rdfs:label>ArgOnt_Instance_50488</rdfs:label>
  <kb:edgeFromINode rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_50485"/>
  <kb:supports rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_50485"/>
  <kb:fulfilsPremiseDesc rdf:resource="http://protege.stanford.edu/kb#ArgOnt_Instance_6"/>
</rdf:Description>
```

References

[1] G. Antoniou, F. van Harmelen, A Semantic Web Primer (Cooperative Information Systems), MIT Press, Cambridge MA, USA, 2004.

[2] K. Atkinson, T. Bench-Capon, P. McBurney, PARMENIDES: facilitating deliberation in democracies, Artificial Intelligence and Law (2006) online only.

[3] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. Patel-Schneider (eds.), The Description Logic Handbook, Cambridge University Press, Cambridge, UK, 2003.

[4] T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web, Scientific American (2001) 29–37.

[5] D. Brickley, R. V. Guha, RDF vocabulary description language 1.0: RDF Schema, W3C Recommendation REC-rdf-schema-20040210, World Wide Web Consortium (W3C) (February 2004).
URL `http://www.w3.org/TR/rdf-schema/`

[6] J. Broekstra, F. van Harmelen, A. Kampman, Sesame: A generic architecture for storing and querying RDF and RDF Schema, in: I. Horrocks, J. A. Hendler (eds.), Proceedings of the First International Semantic Web Conference, ISWC 2002, vol. 2342 of Lecture Notes in Computer Science, Springer-Verlag, 2002.

[7] S. Buckingham Shum, V. Uren, G. Li, B. Sereno, C. Mancini, Modelling naturalistic argumentation in research literatures: Representation and interaction design issues, International Journal of Intelligent Systems, Special Issue on Computational Modelling of Naturalistic Argumentation 22 (1) (2007) 17–47.

[8] D. Carbogim, D. Robertson, J. Lee, Argument-based applications to knowledge engineering, Knowledge Engineering Review 15 (2) (2000) 119–149.

[9] S. Cayzer, Semantic blogging and decentralized knowledge management, Communications of the ACM 47 (12) (2004) 47–52.

[10] C. I. Chesñevar, A. Maguitman, R. Loui, Logical models of argument, ACM Computing Surveys 32 (4) (2000) 337–383.

[11] C. I. Chesñevar, J. McGinnis, S. Modgil, I. Rahwan, C. Reed, G. Simari, M. South, G. Vreeswijk, S. Willmott, Towards an argument interchange format, The Knowledge Engineering Review 21 (4) (2007) 293–316.

[12] J. Conklin, Designing organizational memory: Preserving intellectual assets in a knowledge economy, Gognexus White Paper (2001).

[13] J. Conklin, M. L. Begeman, gIBIS: a hypertext tool for exploratory policy discussion, ACM transactions on office information systems 6 (4) (1988) 303–331.

[14] T. Davies, B. O'Connor, A. A. Cochran, J. J. Effrat, An online environment for democratic deliberation: Motivations, principles, and design, Working paper, Symbolic Systems Program, Stanford University (March 2004).

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

[16] S. Farnham, H. R. Chesley, D. E. McGhee, R. Kawal, J. Landau, Structured online interactions: improving the decision-making of small discussion groups, in: CSCW '00: Proceedings of the 2000 ACM conference on Computer supported cooperative work, ACM Press, New York, NY, USA, 2000.

[17] T. V. Gelder, A Reason!Able approach to critical thinking, Principal Matters: The Journal for Australasian Secondary School Leaders 34 (6) (2002) 34–39.

[18] T. F. Gordon, An open, scalable and distributed platform for public discourse, in: K. Dittrich, W. König, A. Oberweis, K. Rannenberg, W. Wahlster (eds.), Informatik 2003: Innovative Informatikanwendungen, vol. 2 of Lecture Notes in Computer Science, Springer Verlag, Berlin, Germany, 2003, pp. 232–234.

[19] T. F. Gordon, N. Karacapilidis, The Zeno argumentation framework, in: Proceedings of the Sixth International Conference on AI and Law, ACM Press, New York, NY, USA, 1997.

[20] T. F. Gordon, D. Walton, The Carneades argumentation framework, in: P. Dunne, T. Bench-Capon (eds.), Proceedings of the 1st International Conference on Computational Models of Argument (COMMA), IOS Press, Amsterdam, Nethrelands, 2006.

[21] W. S. Grennan, Informal Logic: Issues and Techniques, McGill Queens University Press, Montreal, Quebec, Canada, 1997.

[22] D. Hitchcock, Sampling scholarly arguments: a test of a theory of good inference, in: H. V.Hansen, C. W. Tindale (eds.), Argument and its applications: Proceedings of OSSA 2005, 2001.

[23] N. Karacapilidis, D. Papadias, Computer supported argumentation and collaborative decision making: the HERMES system, Information Systems 26 (2001) 259–277.

[24] G. Karvounarakis, A. Magkanaraki, S. Alexaki, V. Christophides, D. Plexousakis, M. Scholl, K. Tolle, Querying the Semantic Web with RQL, Computer Networks 42 (5) (2003) 617–640.

[25] P. A. Kirschner, S. J. B. Schum, C. S. Carr (eds.), Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making, Springer Verlag, London, 2003.

[26] G. Klyne, J. J. Carroll, Resource description framework (RDF): Concepts and abstract syntax, W3C Recommendation REC-rdf-concepts-20040210, World Wide Web Consortium (W3C) (February 2004).
URL `http://www.w3.org/TR/rdf-schema/`

[27] J. Lee, SIBYL: a tool for managing group decision rationale, in: Proceedings of the conference on computer-supported cooperative work, ACM Press, New York, USA, 1990.

[28] B. Leuf, W. Cunningham, The Wiki Way: Quick collaboration on the Web, Addison-Wesley Professional, Reading MA, USA, 2001.

[29] R. Lüehrs, T. Malsch, K. Voss, Internet, discourses and democracy, in: T. Terano, T. Nishida, A. Namatame, S. Tsumoto, Y. Ohsawa, T. Washio (eds.), New Frontiers in Artificial Intelligence, vol. 2253 of Lecture Notes in Computer Science, Springer-Verlag, Heidelberg, Germany, 2001, pp. 67–74.

[30] P. McBurney, S. Parsons, Risk agoras: Dialectical argumentation for scientific reasoning, in: C. Boutilier, M. Goldszmidt (eds.), Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI-2000), Morgan Kaufmann, Stanford CA, USA, 2000.

[31] P. McBurney, S. Parsons, Representing epistemic uncertainty by means of dialectical argumentation, Annals of Mathematics and Artificial Intelligence, Special Issue on Representations of Uncertainty 32 (1–4) (2001) 125–169.

[32] D. L. McGuinness, Question answering on the Semantic Web, IEEE Intelligent Systems 19 (1) (2004) 82–85.

[33] D. L. McGuinness, F. van Harmelen, OWL web ontology language overview, W3C Recommendation REC-owl-features-20040210/, World Wide Web Consortium (W3C) (February 2004).
URL `http://www.w3.org/TR/owl-features/`

[34] N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Fergerson, M. A. Musen, Creating Semantic Web contents with Protégé-2000, IEEE Intelligent Systems 16 (2) (2001) 60–71.

[35] C. Perelman, L. Olbrechts-Tyteca, The New Rhetoric: a treatise on argumentation, University of Notre Dame Press, 1969.

[36] H. Prakken, C. Reed, D. N. Walton, Dialogues about the burden of proof, in: Proceedings of the 10th international conference on artificial intelligence and law (ICAIL), ACM Press, New York NY, USA, 2005.

[37] H. Prakken, G. Vreeswijk, Logics for defeasible argumentation, in: D. Gabbay, F. Guenthner (eds.), Handbook of Philosophical Logic, vol. 4, 2nd ed., Kluwer Academic Publishers, Dordrecht, Netherlands, 2002, pp. 219–318.

[38] I. Rahwan, Guest Editorial: Argumentation in Multi-Agent Systems, Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS) 11 (2) (2005) 115–125.

[39] I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons, L. Sonenberg, Argumentation based negotiation, Knowledge Engineering Review 18 (4) (2003) 343–375.

[40] I. Rahwan, P. V. Sakeer, Towards representing and querying arguments on the semantic web, in: P. Dunne, T. Bench-Capon (eds.), Proceedings of the 1st International Conference on Computational Models of Argument (COMMA), IOS Press, Amsterdam, Nethrelands, 2006.

[41] C. Reed, Preliminary results from an argument corpus, in: E. M. Bermúdez, L. R. Miyares (eds.), Linguistics in the Twenty First Century, Cambridge Scholars Press, 2006.

[42] C. Reed, J. Katzav, On argumentation schemes and the natural classification of arguments, Argumentation 18 (2) (2004) 239–259.

[43] C. Reed, T. J. Norman (eds.), Argumentation Machines: New Frontiers in Argument and Computation, vol. 9 of Argumentation Library, Kluwer Academic Publishers, Dordrecht, Nethrelands, 2004.

[44] C. Reed, G. Rowe, Translating Toulmin diagrams: Theory neutrality in argument representation, Argumentation 19 (2005) 267–286.

[45] G. W. A. Rowe, C. A. Reed, J. Katzav, Araucaria: Marking up argument, in: European Conference on Computing and Philosophy, 2003.

[46] T. Salminen, M. Marttunen, L. Laurinen, Grounding and counter-argumentation during face-to-face and synchronous network debates in secondary school, in: F. H. van Eemeren, J. A. Blair, C. A. Willard, A. F. Snoeck Henkemans (eds.), Proceedings of the 5th International Conference of the International Society for the Study of Argumentation (ISSA), 2002.

[47] A. F. Snoeck Henkemans, Argument or explanation? propositional relations as clues for distinguishing arguments from explanations, in: F. H. van Eemeren, R. Grootendorst, J. A. Blair, C. A. Willard (eds.), Proceedings of the 4th International Conference of the International Society for the Study of Argumentation (ISSA), 1998.

[48] A. F. Snoeck Henkemans, Indicators of analogy argumentation, in: F. H. van Eemeren, J. A. Blair, C. A. Willard, A. F. Snoeck Henkemans (eds.), Proceedings of the 5th International Conference of the International Society for the Study of Argumentation (ISSA), 2002.

[49] J. F. Sowa, Semantic networks, in: S. C. Shapiro (ed.), Encyclopedia of Artificial Intelligence, John Wiley & Sons, New York, USA, 1992.

[50] S. Toulmin, The Uses of Argument, Cambridge University Press, Cambridge, UK, 1958.

[51] V. Uren, S. Buckingham Shum, M. Bachler, G. Li, Sensemaking tools for understanding research literatures: Design, implementation and user evaluation, International Journal of Human Computer Studies 64 (5) (2006) 420–445.

[52] F. H. van Eemeren, R. F. Grootendorst, Argumentation, Communication and Fallacies: A Pragma-Dialectical Perspective, Erlbaum, USA, 1992.

[53] F. H. van Eemeren, R. F. Grootendorst, F. S. Henkemans, Fundamentals of Argumentation Theory: A Handbook of Historical Backgrounds and Contemporary Applications, Lawrence Erlbaum Associates, Hillsdale NJ, USA, 1996.

[54] B. Verheij, Artificial argument assistants for defeasible argumentation, Artificial Intelligence 150 (1–2) (2003) 291–324.

[55] B. Verheij, Virtual Arguments: On the Design of Argument Assistants for Lawyers and Other Arguers, T.M.C. Asser Press, The Hague, Nethrelands, 2005.

[56] M. Völkel, M. Krötzsch, D. Vrandecic, H. Haller, R. Studer, Semantic wikipedia., in: L. Carr, D. D. Roure, A. Iyengar, C. A. Goble, M. Dahlin (eds.), Proceedings of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 23-26, 2006, ACM Press, New York, NY, USA, 2006.

[57] D. N. Walton, Argumentation Schemes for Presumptive Reasoning, Erlbaum, Mahwah NJ, USA, 1996.