Towards Large Scale Argumentation Support on the Semantic Web

Iyad Rahwan
Institute of Informatics
British University in Dubai
P.O.Box 502216, Dubai, UAE
(Fellow) School of Informatics
University of Edinburgh, UK

Fouad Zablith
Institute of Informatics
British University in Dubai
P.O.Box 502216, Dubai, UAE

Chris Reed
School of Computing
University of Dundee, UK
Dundee DD1 4HN, Scotland

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 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. Users can also attack or support parts of existing arguments, use existing parts of an argument in the creation of new arguments, or create new argumentation schemes. As such, 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.

Introduction

A variety of opinions and arguments are presented every day on the Web, in discussion forums, blogs, 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.

Imagine querying the Web by asking '*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 based on strength (calculated based on 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 news 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 (Chesñevar *et al.* 2007), and capturing Walton's general theoretical account of argumentation schemes (Walton 1996). For the software foundation, we implement the ontology using the RDF Schema ontology language (Brickley & Guha 2004) and present a pilot Semantic Web-based system, ArgDF, through which users can create arguments using different schemes and can query arguments using a Semantic Web query language. Users can also 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 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.

The 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. Finally, the paper contributes to the recently proposed Argument Interchange Format (AIF) ontology (Chesñevar *et al.* 2007) by extending it to capture Walton's argument schemes (Walton 1996) and providing a complete implementation of the AIF in a Semantic Web language. 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.

Related Work

Argumentation-based techniques have found a wide range of applications in artificial intelligence and computer science. An area that has witnessed significant growth is *argumentation-support systems* (ASS) (Kirschner, Shum, & Carr 2003). State-of-the-art ASS's suffer two main limitations. Firstly, they usually support a small number of participants. Secondly, most of them target specific domains, such as education (Rowe, Reed, & Katzav 2003) or jurisprudence ArguMed (Verheij 2003). Consequently, they are based on specialised approaches to argumentation.

The World Wide Web can be seen as an ideal platform for supporting large-scale argumentative expression and communication, due to its ubiquity and openness. On-line discussion forums, such as *Deme* (Davies *et al.* 2004), can provide a medium for such communication. However, while discussions may be identified by their topics, time, or participants, there is a lack of fine-grained structure that captures the details of how different facts, opinions, and arguments contribute to the overall result. Having such structure could enable better visualisation, navigation and analysis of the 'state of the debate' by participants or automated tools. Automate support for argumentation may exploit the underlying structure, for example, to discover inconsistencies or synergies among arguments.

Recently, some Web-based tools have begun to enable simple structuring of arguments. The *truthmapping*[1] system supports a large number of participants but it only 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*,[2] which was released to the public in late 2006. It provides a forum for commentators to post their opinions about political events and issues. 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* (Völkel *et al.* 2006) and can export content into RDF format for use by other Semantic Web applications.

Our aim in this paper is to combine the strengths of ASS's and the Semantic Web to enable highly-scalable yet highly-structured argument representation and processing capability in an open Web environment.

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-size groups of participants, we aim to build an open, extensible and reusable infrastructure for large-scale argument representation, manipulation, and (eventually) evaluation. We now list a set of key requirements that we believe are important to enable 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;
5. The WWAW must support the representation, annotation and creation of arguments using a variety of schemes.

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

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

Argument Interchange Format (AIF): Core

The AIF is a core ontology of argument-related concepts, and can be extended to capture a variety of argumentation formalisms and schemes. The AIF core ontology assumes that argument entities can be represented as nodes in a directed graph called an *argument network*.

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. Moreover, properties and relations between classes and instances (including graph edges) will be captured through predicates over sets. 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, certainty degree, acceptability status, etc.

The core AIF has two types of nodes: *information nodes* (or *I-nodes*) and *scheme nodes* (or *S-nodes*). These are represented by two disjoint sets, $\mathcal{N}_I \subset \mathcal{N}$ and $\mathcal{N}_S \subset \mathcal{N}$, respectively. 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 inference. The schemes themselves belong to a class, $\mathcal{S}$, and are classified into the types: *rule of inference scheme*, *conflict scheme*, and *preference scheme*. 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. 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, etc.), and $\mathcal{N}_S^{PA}$ are applications of (possibly abstract) criteria of preference among evaluated nodes.

The AIF core specification does not type its edges. 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. This ensures that the type of any relationship between two pieces of information must be specified explicitly via an intermediate S-node.

	to *I-node*	to *RA-node*	to *PA-node*	to *CA-node*
from *I-node*		I-node data used in applying an inference	I-node data used in applying a preference	I-node data in conflict with information in node supported by CA-node
from *RA-node*	inferring a conclusion in the form of a claim	inferring a conclusion in the form of an inference application	inferring a conclusion in the form of a preference application	inferring a conclusion in the form of a conflict definition application
from *PA-node*	applying a preference over data in I-node	applying a preference over inference application in RA-node	meta-preferences: applying a preference over preference application in supported PA-node	preference application in supporting PA-node in conflict with preference application in PA-node supported by CA-node
from *CA-node*	applying conflict definition to data in I-node	applying conflict definition to inference application in RA-node	applying conflict definition to preference application in PA-node	showing a conflict holds between a conflict definition and some other piece of information

Table 1: Informal semantics of untyped edges in core AIF

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.*

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 rule of inference application node; 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 a description of a simple argument in propositional logic, depicted graphically in Figure 1(a). We distinguish S-nodes from I-nodes graphically by 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 nodes (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 captures 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 1(b), between two simple arguments.

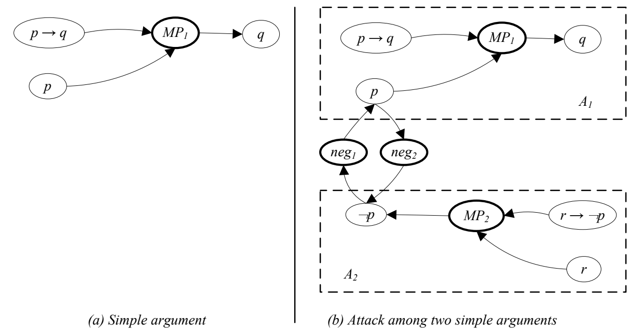

Figure 1: Examples of simple arguments

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.

Extending the Core AIF: Argument Schemes

Argumentation schemes are *forms* of argument, representing stereotypical ways of drawing inferences from particular patterns of premises and conclusions. Schemes help categorise the way arguments are built. Among others, Walton's taxonomy (Walton 1996) has 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. A common example of Walton-style schemes is the *Argument from Expert Opinion*:

– **Premise:** Source E is an expert in the subject domain S.

– **Premise:** E asserts that A, in domain S, is true.

– **Conclusion:** A may plausibly be taken to be true.

Many other schemes were presented by Walton (Walton 1996), 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 3 ***(Instance of Argument from Expert Opinion)***

– **Premise:** *Allen is an expert in sport.*

– **Premise:** *Allen says that Brazil has the best football team.*

– **Conclusion:** *Brazil has the best football team.*

Critical questions serve to inspect arguments based on this scheme. For example, in the canonical scheme for "Argument from expert opinion," there are six critical questions:

1. *Expertise Question:* How credible is expert E?

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 Gordon & Walton in the Carneades model (Gordon & Walton 2006), these questions are not all alike. The first, second, third and sixth questions refer to *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. Numbers four and five, however, shift the burden of proof to the questioner (e.g., the opponent must demonstrate that another expert disagrees with E). These questions capture *exceptions* to the rule, and correspond to Toulmin's *rebuttal* (Toulmin 1958).

Recall that in Example 1, we represented the rule of inference application in an RA-node labelled MP_1, and stated explicitly that it uses the modus ponens generic natural deduction rule. It would therefore seem natural to use the same approach with presumptive schemes. However, this approach loses the information about the generic structure of the scheme, as well as 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 this end, we propose capturing the structure of the scheme explicitly in the argument network.

We consider the set of schemes $\mathcal{S}$ as nodes in the argument network. And we introduce a new class of nodes, called *forms* (or *F-nodes*), captured in the set $\mathcal{N}_F \subseteq \mathcal{N}$. Two distinct types of forms are presented: *premise descriptors* and *conclusion descriptors*, denoted by $\mathcal{N}_F^{Prem} \subseteq \mathcal{N}_F$ and $\mathcal{N}_F^{Conc} \subseteq \mathcal{N}_F$, respectively. As can be seen in Figure 2, we can now explicitly link each node in the actual argument (the four unshaded nodes at the bottom right) to the form node it instantiates (the four shaded nodes at the top right).[3] Notice that here, we expressed the predicate 'uses' with the edge $\xrightarrow{fulfilsScheme}: \mathcal{N}_S \times \mathcal{S}$.

Since each critical question corresponds either to a presumption or an exception, we provide explicit descriptions of the presumptions and exceptions associated with each scheme. To express the scheme's presumptions, we add a new type of F-node called *presumption*, represented by the set $\mathcal{N}_F^{Pres} \subseteq \mathcal{N}_F$, and linked to the scheme via a new edge type $\xrightarrow{hasPresumption}: \mathcal{S} \times \mathcal{N}_F^{Pres}$. This is shown in the three (shaded) presumption nodes at the bottom left of Figure 2. As for representing exceptions, the AIF offers a more expressive possibility. 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, 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 be presented as conflict scheme descriptions (as shown in the top left of Figure 2).

Note that now, there is no longer any need to represent critical questions directly in the network, since they are easily 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?*'

Finally, in Walton's account of schemes, some presumptions 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. This relationship, between some premises and presumptions, can be captured explicitly using a predicate ($\xrightarrow{entails}: \mathcal{N}_F^{Prem} \times \mathcal{N}_F^{Pres}$).

Definition 3 ***(Presumptive Inference Scheme Description)*** *A* presumptive inference scheme description *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*
- $\xrightarrow{entails} \subseteq \mathcal{N}_F^{Prem} \times \mathcal{N}_F^{Pres}$

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 rule of inference application node; 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.*

[3] To improve readability, we will start using typed edges, which will enable us to explicitly distinguish between the different types of connections between nodes. All typed edges will take the form $\xrightarrow{type}$, where $type$ is the type of edge, and $\xrightarrow{type} \subseteq \xrightarrow{edge}$.

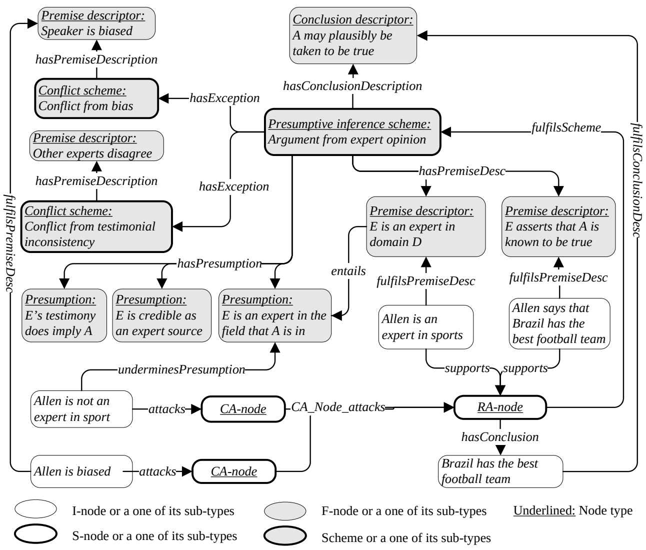

Figure 2: An argument network showing an argument from expert opinion, two counter-arguments undermining a presumption and an exception, and the descriptions of the schemes used by the argument and attackers. *A:* Brazil has the best football team: Allen is a sports expert and he says so; *B:* But Allen is biased, and he is not an expert in sports!

ArgDF: A Semantic Web System for Authoring and Navigating Arguments

We implemented our extended ontology using RDF and RDFS,[4] and call the resulting ontology AIF-RDF. In summary, we view elements of arguments and schemes (e.g. premises, conclusions) as RDF resources, and connect them using binary predicates as described earlier.

ArgDF[5] is a Semantic Web-based system built on the top of the AIF-RDF ontology. It uses a variety of software components such as the Sesame RDF repository,[6] PHP scripting, XSLT, the Apache Tomcat server,[7] and MySQL database. The system also uses Phesame,[8] 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 the Semantic Web query language RQL; and (iv) returning RDF query results in a variety of computer processable formats including XML, HTML or RDF.

Creating New Arguments: The system presents the available schemes, and allows the user to choose the scheme to which the argument belongs. Details of the selected scheme are then retrieved from the repository, and the form of the argument is displayed to the user, who then creates the conclusion followed by the premises.

Support/Attack of Existing Expressions: The list of existing expressions (i.e. premises or conclusions) in the repository can be displayed. The user can choose an expression to support or attack. When a user chooses to support an existing premise through a new argument/scheme, this premise will be both a premise in one argument, and a conclusion in another. Thus, the system enables argument *chaining*. If the user chooses to *attack* an expression, on the other hand, s/he will be redirected to choose an appropriate conflict scheme, and create a new argument whose conclusion is linked to the existing conclusion via a conflict application node (as in Example 2).

Searching through Arguments: The system also enables users to search existing arguments, by specifying text found in the premises or the conclusion, the type of relationship between these two (i.e. support or attack), and the scheme(s) used. For example, one can search for arguments, based on expert opinion, *against* the '*war on Iraq*,' and mentioning '*weapons of mass destruction*' in their premises. In the background, the system construct an RQL query which is then submitted to the RDF repository.

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 premise thus contributes to multiple arguments in a *divergent* structure. 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.

Attacking Arguments through Implicit Assumptions: With our account of presumptions and exceptions, it becomes possible to construct an automatic mechanism for *presuming*. ArgDF allows the user to inspect an existing argument, allowing the exploration of the hidden assumptions (i.e. presumptions and exceptions) by which its inference is warranted. This leads the way for possible implicit attacks on the argument through pointing out an exception, or through undermining one of its presumptions (as shown in Figure 2). This is exactly the role that Walton envisaged for his critical questions (Walton 1996). Thus, ArgDF exploits knowledge about implicit assumptions in order to enable richer interaction between the user and the arguments.

Creation of New Schemes: The user can create new schemes through the interface of ArgDF without having to modify the ontology.[9] This feature enables a variety of user-created schemes to be incorporated, thus offering flexibility not found in any other argument-support system.

---

[4]The Resource Description Framework (RDF) is an XML-based language for making statements about resources. Each resource has a Universal Resource Identifier (URI). A statement is a subject-predicate-object expression, sometimes called a *triple*. The predicate captures a relationship between the subject (a resource) and the object (another resource, or a *literal*). RDF Schema (RDFS) (Brickley & Guha 2004) is a Semantic Web ontology language. It provides constructs for specifying classes and class hierarchies, properties (or predicates) and property hierarchies, restrictions on the domains and ranges of properties, etc. RDFS specifications are themselves RDF statements. For example, the triple (`I-Node, rdfs:subClassOf, Node`) specifies that '`I-Node`' is a sub-class of '`Node`.'

[5]ArgDF is currently a proof-of-concept prototype and can be accessed at: `http://www.argdf.org`

[6]See: `http://www.openrdf.org`

[7]See: `http://tomcat.apache.org/`

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

[9]Recall that actual schemes are instances of the "Scheme" class.

Conclusions and Future Possibilities

As tools for electronic argumentation grow in sophistication, number and popularity, so the role for the AIF and its implementations will 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 and the Semantic Web. Earlier in the paper, we introduced desiderata necessary for the creation of a WWAW and we conclude here by revisiting them.

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 an RDF repository which can be accessed and queried openly through the Web and using a variety of standard RDF query languages.
3. *The WWAW language must be based on open standards, enabling collaborative development of new tools:* Arguments in ArgDF are annotated in RDF using RDFS ontologies, both open standards endorsed by the W3C.
4. *The WWAW must employ a unified, extensible argumentation ontology:* Our ontology captures the main concepts in the Argument Interchange Format ontology (Chesñevar *et al.* 2007), 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.

AIF represents a first step towards an open mechanism for representing arguments, but the high level of abstraction that was demanded of it also presents challenges to developers' abilities to use it. AIF-RDF bridges this gap between the ontological abstraction and the code-level detail. ArgDF then demonstrates the flexibility that AIF-RDF affords, and offers an example of rapid tool development on the basis of theoretical advances in the understanding of argument structure. Following are some potential usage scenarios that may exploit the infrastructure presented here.

Question Answering: One extension of the current system is to exploit the variety of Semantic Web techniques for improving question answering (McGuinness 2004). Prospects range from using query refinement techniques to interactively assist users find arguments of interest through Web-based forms, to processing natural language questions to generate queries. This functionality would require annotations of a large amount content on the Web. Translating the ontology to more expressive Semantic Web ontology languages such as OWL (McGuinness & van Harmelen 2004) can also enable ontological reasoning over argument structures, for example, to automatically classify arguments, or to identify semantic similarities among them.

Interface and argument visualisation: ArgDF provides only rudimentary displays. More intuitive argument visualisation is needed for the WWAW to appeal to non-experts.

Argumentative Blogging: Another potential extension is combining our framework with so-called *Semantic Blogging* tools (Cayzer 2004), 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 potentially help build up large amounts of annotations, making the question answering scenario above more viable.

Mass-collaborative argument editing: Accumulating argument annotations can be done through mass-collaborative editing of semantically connected argumentative documents in the style of *Semantic Wikipedia* (Völkel *et al.* 2006). A basic feature of this kind is already offered by Discourse DB.

References

Brickley, D., and Guha, R. V. 2004. RDF vocabulary description language 1.0: RDF Schema. W3C Recommendation REC-rdf-schema-20040210, W3C.

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

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

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

Gordon, T. F., and Walton, D. 2006. The Carneades argumentation framework. In *Proc. 1st Intl. Conf. on Computational Models of Argument*, 195–207. IOS Press.

Kirschner, P. A.; Shum, S. J. B.; and Carr, C. S., eds. 2003. *Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making*. London: Springer.

McGuinness, D. L., and van Harmelen, F. 2004. OWL web ontology language overview. W3C Recommendation REC-owl-features-20040210, W3C.

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

Rowe, G. W. A.; Reed, C. A.; and Katzav, J. 2003. Araucaria: Marking up argument. In *European Conference on Computing and Philosophy*.

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

Verheij, B. 2003. Artificial argument assistants for defeasible argumentation. *Artificial Intelligence* 150(1-2):291–324.

Völkel, M.; Krötzsch, M.; Vrandecic, D.; Haller, H.; and Studer, R. 2006. Semantic wikipedia. In *Proc. WWW 2006, Edinburgh, UK*, 585–594. ACM Press.

Walton, D. N. 1996. *Argumentation Schemes for Presumptive Reasoning*. Mahwah NJ, USA: Erlbaum.