Mass Argumentation and the Semantic Web

Iyad Rahwan [a,b]

[a] *(Fellow) School of Informatics, University of Edinburgh, Edinburgh EH8 9LE, UK*
[b] *Institute of Informatics, British University in Dubai, P.O.Box 502216, Dubai, UAE*

Abstract

The World Wide Web (WWW) can be seen as an ideal platform for enhancing argumentative expression and communication, due to its ubiquity and openness. Much argumentation takes place on personal blogs and on unstructured or semi-structured discussion forums. Recently, an increasing number of Web 2.0 applications provide specific support for large-scale socially-contributed argumentative content. When compared with traditional methods of Web discourse, these tools enable better visualisation, navigation and analysis of the 'state of the debate' by participants and, potentially, by automated tools. In this paper, I outline some potential benefits of Semantic Web techniques in supporting mass-scale, socially-contributed argument tagging. I also present some recent research in this direction.

Key words: Argumentation, Web 2.0, Semantic Web

1. Argumentation Theory: A Crash Course

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 intended to justify (or refute) the standpoint before a rational judge*" [1, page 5]. In the next subsections, I outline some key concepts in describing and evaluating arguments and argumentative processes.

1.1. *Elements of Argument*

Superficially, an argument consists of a *conclusion*, which is a claim made by the arguer, and a set of *premises* that support the conclusion.

Example 1 ***(Argument)***

– ***Premise:*** *If Iraq has weapons of mass destruction (WMDs), then it poses a threat to world peace;*
– ***Premise:*** *Iraq has WMDs;*
– ***Conclusion:*** *Iraq poses a threat to world peace.*

Walton [2] identifies three major types of argument. In a *deductive argument* (e.g. mathematical proof in propositional logic), if the premises are true, then the conclusion is necessarily true. An *inductive argument* involves a kind of generalisation from gathered empirical evidence. Inductive arguments sometimes use statistical techniques to establish the strength (or confidence) of the supported claim. Finally, in a *presumptive argument*, the conclusions are said to be plausible given the premises. Plausibility is different from probability. While probability is determined by reasoning from statistical evidence, plausibility states that the conclusion holds by default provided no sufficient evidence supports the contrary. Arguments can be depicted graphically using *argument diagramming* techniques. Figure 1 shows a simple graphical presentation of a deductive argument in propositional logic.

Email address: irahwan@acm.org (Iyad Rahwan).
URL: http://homepages.inf.ed.ac.uk/irahwan/ (Iyad Rahwan).

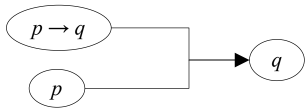

Fig. 1. A simple argument

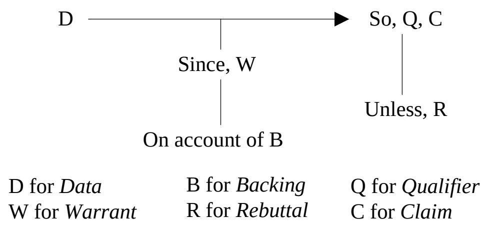

Fig. 2. The general form of Toulmin's model of argument

Other views of arguments –beyond the simple distinction between premises and conclusions– have been proposed by argumentation theorists to aid the analysis of argumentative prose or discourse. A notable model of argument is that presented by Toulmin [3]. Toulmin proposed that an argument can be usefully analysed using six interrelated components:

(i) **Claim:** a conclusion that is being established. E.g. "*Iraq poses a threat to world peace.*"
(ii) **Data:** the facts we appeal to in order to support the claim. E.g. "*Iraq has WMDs.*"
(iii) **Warrant:** a statement authorizing concluding the claim from the data. E.g. "*Countries that have WMDs pose a threat to world peace.*"
(iv) **Backing:** credentials that certify the statement expressed in the warrant. E.g. one may back the above warrant by citing historical evidence that countries with WMDs have indeed disrupted world peace.
(v) **Rebuttal:** statements recognising the restrictions to which the claim may legitimately be applied. E.g. "*Countries that have WMDs pose a threat to world peace, unless these countries are democracies.*" Rebuttals, in a way, provide opportunities to *attack* the argument.
(vi) **Qualifier:** words expressing the degree of force or certainty concerning the claim. Examples of such words include "*possible,*" "*certainly,*" "*necessarily,*" "*presumably,*" etc.

The general form of Toulmin's model is depicted in Figure 2 [3, page 104].

More recently, there has been increasing interest in classifying different types (or schemes) of arguments based on generic types of premises and conclusions they satisfy. 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 [4], and Katzav and Reed [5]. But it is Walton's exposition [6] that has been most influential in computational work. Each *Walton scheme* has a name, conclusion, set of premises and a set of critical questions. 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 2 ***(Scheme for Argument from Expert Opinion)***
– **Premise:** *Source E is an expert in domain S.*
– **Premise:** *E asserts that proposition A is true.*
– **Conclusion:** *A may plausibly be taken to be true.*

Other schemes include *argument from consequence*, and *argument from analogy*, etc. Actual arguments are *instances* of schemes.

Example 3 ***(Instance of Argument from Expert Opinion)***
– ***Premise:*** *The CIA say that Iraq has WMDs.*
– ***Premise:*** *The CIA are experts in WMDs.*
– ***Conclusion:*** *Iraq has WMDs.*

Note that premises may not always be stated, in which case we say that a given premise is *implicit* [2] or *unexpressed* [7]. One of the benefits of argument classification is that it enables analysts to uncover the hidden premises behind an argument, once the scheme has been identified.

1.2. *Combining Arguments*

The abstract arguments presented in the previous sections are referred to as *linked arguments* [2], since they link a set of premises which, together, lead to the conclusion. A special case of linked arguments are *single arguments* [2], which include only a single premise. Single and linked arguments are shown in Figure 3(i) and 3(ii).

It is often the case that arguers present complex structures of multiple inter-connected arguments. For example, one may present multiple (individual) arguments in support of the same conclusion. The resulting argument structure is referred to as *conver-*

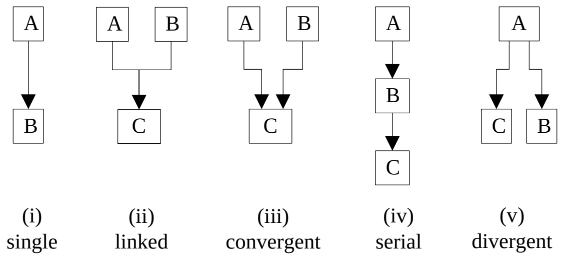

Fig. 3. Common basic argument structures

gent argument [2] (also referred to as *multiple argumentation* [7]), and is depicted in Figure 3(iii). Note the difference between linked and convergent arguments: in convergent arguments, each premise alone supports the conclusion (i.e., together with the conclusion, it constitutes a single argument by itself). Another common structure is the *serial argument* [2] (also referred to as *subordinative argumentation* [7]). In this structure, the conclusion of one argument acts as a premise of another, whose conclusion could also form a premise of another argument, and so on, thus resulting in a chain of arguments as shown in Figure 3(iv). A third important structure is the *divergent argument* [2], in which a single statement acts as a premise to support multiple conclusions. This kind of structure is depicted in Figure 3(v). Note that combinations of these structures are also possible, resulting in more complex arguments.

1.3. *Argument Evaluation*

The previous section outlined some major trends in the *description* of argument structures as well as argumentative dialogues. This section outlines key ideas in the *evaluation* of argumentative statements as well as argumentative discourse.

One way to evaluate arguments is through *critical questions*, which serve to inspect arguments based on a particular argument scheme. For example, Walton [6] identified six critical questions for "Argument from expert opinion:"

(i) *Expertise:* How credible is expert E?
(ii) *Field:* Is E an expert in the field that the assertion, A, is in?
(iii) *Opinion:* Does E's testimony imply A?
(iv) *Trustworthiness:* Is E reliable?
(v) *Consistency:* Is A consistent with the testimony of other experts?
(vi) *Backup Evidence:* Is A supported by evidence?

As discussed by Gordon and Walton in the Carneades model [8], 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* [3].

In computer science, much interest is dedicated to identifying criteria for *argument acceptability* based on the complex chains of attacks. In his seminal work, Dung [9] views arguments as abstract entities with an *attack* (or *defeat*) relation among them. Formally, an argumentation framework is defined as a pair $AF = \langle \mathcal{A}, \rightarrow \rangle$, where $\mathcal{A}$ is a set of arguments and $\rightarrow$ is an attack relation. This corresponds to an argument graph that characterises the conflicts among arguments. With this structure in place, it becomes possible to present various criteria for characterising the possible sets of *acceptable* arguments. For example, under sceptical acceptability, an argument A_1 is acceptable if it is not attacked, or if is *defended* against each of its attackers A' by an acceptable argument A_2 (i.e., A_2 is acceptable and attacks A'). Under credulous acceptability, an argument is acceptable if it is part of a maximal admissible (i.e. self-defending) set of arguments.

In terms of evaluating argumentation dialogues, a key question is to identify fallacies made by dialogue participants. One of the most formally precise ways of studying dialogues is through *dialogue-games* [10], which are interactions between two or more players, where each player makes a *move* by making some utterance in a common communication language, and according to some pre-defined rules.

2. Arguing on Today's Web

The Web can be seen as an ideal platform for enhancing argumentative expression, due to its ubiquity and openness. 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 structure of argumentative viewpoints explicitly. This makes the task of searching, evaluating, comparing and identifying the relationships among arguments difficult. Recently, a number of Web 2.0 tools have begun to provide more explicit support for argumentation, thus enabling a more explicit structuring of arguments. Below, I describe some of these tools briefly, outlining their key features and limitations.

Truthmapping[1] is a public argumentation support system that supports a large number of participants. It enables users to create arguments that consist of premises and conclusions. Arguments can be chained, and can contain hyperlinks to other Web addresses or to premises or conclusions of other arguments (within or outside the given argument). Moreover, each user can present a single *critique* to each premise. And the person asserting the argument can add a single rebuttal to each critique.

Debatabase[2] is a database of arguments organised into topics, each of which is a (usually yes/no) question. Each question is supplied with a context, and a number of arguments pro and con that question are listed. The database is intended as a resource for students interested in learning debating skills. The topics are written by expert debaters, judges and coaches, and included are background summaries, links to websites of interest and recommended books, example motions and user comments. At the time of writing, there are a total of 423 questions in the database. A similar approach has been taken by *Debatepedia*.[3]

Standpoint[4] is a Web-based system that enables a user to make a claim, and then add links to resources (e.g. blogs, websites, and books) that support this claim. Once a claim is posted, any user can post reasons why they agree with such claim, or what they believe instead of it. A user can also rate the evidence that others have posted. One of the aims of the Web site is to enable users to find people who share similar points of view.

Standpedia[5] is wiki-style encyclopedia of controversy. Standpedia users work together to build maps of controversial issues that everyone can agree on – not because the map offers one right answer but because all the perspectives are well-represented within the map.

[1] See `http://www.truthmapping.com`
[2] `http://www.idebate.org/debatabase/`
[3] `http://www.debatepedia.com`
[4] `http://www.standpoint.com`
[5] `http://www.standpedia.com`

3. Arguing on the Semantic Web

Existing Web 2.0 systems for supporting mass argumentation have a number of limitations. Firstly, there is limited or no integration between the different argument repositories. This limits the ability to provide services (e.g. question answering systems) that make use of arguments from multiple mass argumentation repositories.

Another, related limitation of existing systems is that argument structure is relatively shallow. Most Web 2.0 applications distinguish only between premises and conclusions, and possibly between pro- and con- arguments. But they do not distinguish between different types of arguments, or subtle types of attack among arguments. Moreover, existing tools do not provide semantically rich links among arguments. For example, in truthmapping, while user contributed text (i.e. premises, conclusions, critiques and rebuttals) can contain hyperlinks to any Web content including supporting material or even other "truthmaps" which does enable cross-referencing among arguments, these cross-references carry no explicit semantics. So, it is not possible to connect multiple arguments explicitly in complex structures (e.g. that contain combinations of convergent arguments, divergent arguments, etc.). This limitation inhibits the possibilities for using meta-data about arguments to enhance the automated search and/or evaluation of arguments on the Web.

In the next sub-sections, I outline some directions in which Semantic Web technologies can enhance mass argumentation on today's Web 2.0.

3.1. *Integrating Mass Argumentation Tools*

One of the main challenges in argumentation support tools on the Web is the lack of a shared ontology (or interchange format) for representing arguments and argumentation. As a result, arguments posted on various sites such as *truthmapping* or *debatabase* cannot be easily integrated. Such integration can allow users of one mass argumentation tool to easily access arguments posted on another tool. It can also allow third-party tools to aggregate content from a variety of mass argumentation tools.

Semantic Web technologies are well placed to facilitate the integration among mass argumentation tools. On one hand, a unified argument description ontology could act as an inter-lingua between the different tools. If Web 2.0 mass argumentation tools can provide access to their content through a common ontology (e.g. in RDF or OWL), then software developers could easily build tools to exchange (e.g. import and export) arguments between tools. On the other hand, if a standard ontology of arguments is not attainable, then ontology mapping tools [11] can potentially provide means for the automatic translation of a variety of argument annotation languages.

The most comprehensive and most recent attempt to provide a general ontology of argument-related concepts is the Argument Interchange Format [12], which I shall return to later in the paper.

3.2. *Rich Argument Ontologies for Question Answering*

Arguably, the key feature of Semantic Web technologies is that they represent Web information in standard, machine-processable formats (e.g. in RDF or OWL). Once this kind of explicit semantic mark-up is in place, a wide variety of applications can exploit mark-up to offer better information services. There is an opportunity for exploiting explicit semantic mark-up of arguments to support mass argumentation on the Web.

Semantic mark-up can enable us to explicitly annotate arguments, and to distinguish between the different components of an argument. For example, one can use an XML document to describe the premises and the conclusion of an argument. This is the approach taken in the Argument Markup Language (AML) used in the argument annotation tools Araucaria [13]. The syntax of AML is specified in a Document Type Definition (DTD) which imposes structural constraints on the form of valid AML documents. AML was primarily produced for use in the Araucaria tool, though has more recently been adopted elsewhere. A corpus of analysed and annotated arguments is available in XML format from the AraucariaDB on-line repository, which can be searched using XPath queries. [6] Search can be filtered by text, analyst, date, argument scheme, etc.

A variety of other argument mark-up languages have been proposed in the context of argument visualisation and mapping tools. *ClaiMaker* and related technologies [14] provide a set of tools for individuals or distributed communities to publish and contest ideas and arguments, as is required in contested domains such as research literatures, intelligence analysis, or public debate. This system makes use of the *ScholOnto* ontology [15], which can express a number of basic reasoning schemes (causality, support) and relationships between concepts found in scholarly discourse (e.g. similarity of ideas, taxonomies of concepts, etc.). Another mark-up language was developed for Compendium, [7] a semantic hypertext concept mapping tool. The Compendium argument ontology enables constructing *Issue Based Information System (IBIS)* networks, in which nodes represent *issues*, *positions* and *arguments* [16]. The Assurance and Safety Case Environment (ASCE) [8] is a graphical and narrative authoring tool for developing and managing assurance cases, safety cases and other complex project documentation. ASCE relies on an ontology for *arguments about safety* based on *claims*, *arguments* and *evidence* [17].

These various attempts at providing argument mark-up languages share a limitation. Each particular language is designed for use with a specific tool (usually for the purpose of facilitating argument visualisation). As a consequence, the semantics of arguments specified using these languages is tightly coupled with particular schemes to be interpreted in a specific tool and according to a specific underlying theory. Thus, for example, arguments in the Compendium concept mapping tool are to be interpreted in relation to a rigorous theory of issue-based information systems.

Semantic Web ontologies can offer a unified ontology for describing and annotating arguments. If Web 2.0 mass argumentation tools can provide access to their content through a common ontology (e.g. in RDF or OWL), then software developers could easily build tools that aggregate content from various tools. An obvious application of such semantic argument mark-up is to exploit the variety of ideas and techniques for improving question answering by exploiting features of the Semantic Web [18]. 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 WMDs.*' Marking-up arguments using more expressive Semantic Web ontology languages such as OWL [19] can also enable

[6] http://araucaria.computing.dundee.ac.uk/search.php

[7] http://www.compendiuminstitute.org/tools/compendium.htm

[8] http://www.adelard.co.uk/software/asce/

ontological reasoning over argument structures, for example, to automatically classify arguments, or to identify semantic similarities among arguments.

3.3. *Argumentative Semantic Blogging*

One of the major challenges in any Semantic Web application is the so-called knowledge acquisition bottleneck. How can we produce a large amount of semantically marked-up arguments? One way is to provide easy tools for annotating arguments as part of existing Web 2.0 tools. A notable emerging type of tools are *Semantic Blogging* tools [20], which enable users to annotate their blog entries with various mark-ups. In the context of mass argumentation, one can conceive of a semantic blogging tool which enables users to annotate their views explicitly as arguments, to enable other users to easily identify them. A primitive form of this feature has been explored to some extent in the 'Standpoint' system described above, where users can publish their claims as RSS feeds, or subscribe to other people's feeds. A natural extension of this is to enable users to link their arguments explicitly to arguments posted in other blogs. Such links can be semantically rich, indicating the type of relationship, be it an endorsement, a rebuttal, etc. In general, semantic blogging 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.

3.4. *Semantic 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* [21]. A basic feature of this kind is already offered by *Discourse DB*,[9] which was released to the public in 2006 and started to accumulate sizable content. Discourse DB is a forum for journalists and commentators to post their opinions about ongoing political events and issues. Positions on certain matters 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 provides facilities to export content into RDF format for use by other Semantic Web applications.

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

4. An Infrastructure for Unified Semantic Argument Annotation

In the previous section, I outlined a number of potential uses of Semantic Web technologies in supporting mass argumentation. Only a few of these applications exist, and when they do, they make relatively basic use of Semantic Web techniques, and do not provide explicit integration of argumentative content using a shared ontology.

In this section, I present a brief description of an ontology which may be used as a seed for a variety of Semantic Web argument annotation tools. The ontology is described in detail in a recent joint paper with other colleagues [22]. It enables the annotation of arguments using RDF, and is theoretically founded on the Argument Interchange Format (AIF) [12] and Walton's general theoretical account of argumentation schemes [6]. I also describe a pilot system built that exploits this ontology.

4.1. *The Argument Interchange Format*

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 views argument components as nodes in a directed graph called an *argument network*.

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. 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

	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

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 has three different types of scheme nodes: *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 specification does not type its edges. The (informal) semantics of edges can be inferred from the types of nodes they connect (see 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.

Definition 1 ***(Argument Network)*** *An* argument network Φ *is a graph consisting of:*
– *a set* $\mathcal{N} = \mathcal{N}_I \cup \mathcal{N}_S$ *of vertices (or* nodes*); and*
– *a binary relation* $\xrightarrow{edge}: \mathcal{N} \times \mathcal{N}$ *representing edges.*
where $\nexists(i, j) \in \xrightarrow{edge}$ *where both* $i \in \mathcal{N}_I$ *and* $j \in \mathcal{N}_I$

A *simple* argument can be represented by linking a

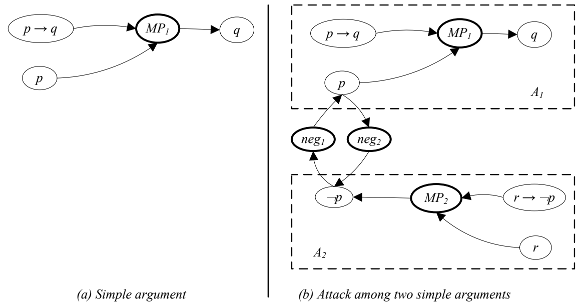

Fig. 4. Examples of simple arguments; S-Nodes denoted with a thicker border

set of premises to a conclusion.

Definition 2 ***(Simple Argument)*** *A* simple argument*, in network* Φ *and schemes* $\mathcal{S}$*, 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;*
– $c \in \mathcal{N}_I$ *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 in Figure 4(a).

Example 4 ***(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 between two simple arguments (see Figure 4(b)).

Example 5 ***(Simple Arguments in Conflict)***
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:* neg_1 *and* neg_2*.*

4.2. *Representing Argument Schemes*

Recall that schemes are *forms* of argument, representing stereotypical ways of drawing inferences from particular patterns of premises to conclusions. 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 5, 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). [10] 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 5. 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). Thus, exceptions can most accurately be presented as conflict scheme descriptions (see top left of Figure 5).

Finally, in Walton's account of schemes, some presumptions may be implicitly or explicitly *entailed* by a premise. 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 is 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;*
- $c \in \mathcal{N}_I$ *is a node denoting the conclusion;*

such that:
- $\tau \xrightarrow{edge} c$*;* $\forall p \in P$ *we have* $p \xrightarrow{edge} \tau$*;*
- $\tau \xrightarrow{fulfilsScheme} \alpha$*;* $c \xrightarrow{fulfilsConclusionDesc} cd$*; and*
- $\xrightarrow{fulfilsPremiseDesc} \subseteq P \times PD$ *corresponds to a one-to-one correspondence from* P *to* PD*.*

4.3. *An Implementation: AIF-RDF and ArgDF*

We implemented our extended ontology using RDF and RDFS [23], 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 [11] is a Semantic Web-based system that uses the AIF-RDF ontology. It uses a variety of software components such as the Sesame RDF reposi-

[10] To improve readability, we will start using typed edges. All typed edges will take the form $\xrightarrow{type}$, where *type* is the type of edge, and $\xrightarrow{type} \subseteq \xrightarrow{edge}$.

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

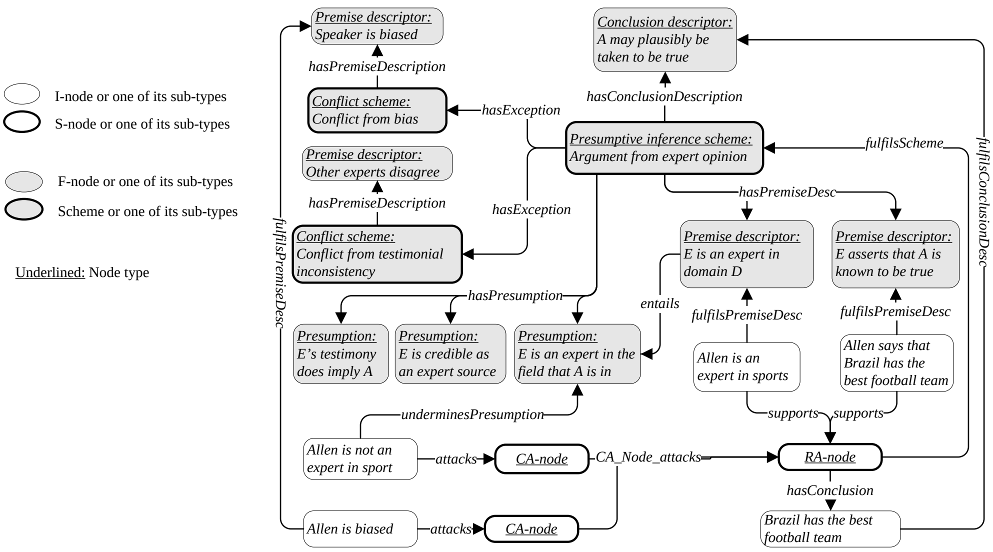

Fig. 5. An argument network showing an argument from expert opinion, two counter-arguments undermining a presumption and an exception, and the descriptions of the schemes used by the argument and attackers. *A:* Brazil has the best football team: Allen is a sports expert and he says so; *B:* But Allen is biased, and he is not an expert in sports!

tory,[12] PHP scripting, XSLT, the Apache Tomcat server,[13] and MySQL database. The system also uses Phesame,[14] 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 expressions (i.e. premises or conclusions) in the repository can be displayed, supported or attacked. 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 5).

Searching through Arguments: The system 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. An RQL query is generated in the background.

Linking Existing Premises to a New Argument: While creating premises supporting a given conclusion through a new argument, the user can *reuse* 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

[12] See: `http://www.openrdf.org`
[13] See: `http://tomcat.apache.org/`
[14] `http://www.hjournal.org/phesame`

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 5). This is exactly the role that Walton envisaged for his critical questions [6]. 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. This feature enables a variety of user-created schemes to be incorporated, thus offering flexibility not found in any other argument-support system.

5. Conclusion

In this paper, I surveyed some existing Web 2.0 applications that support mass-scale, socially-contributed argument tagging. I outlined some potential benefits of Semantic Web techniques in such applications. I then summarised recent work towards a standard argument ontology [12] and its realisation in a Semantic Web pilot system [22]. A major issue of future work is the adoption of the proposed ontology. One would not expect normal Web users to use the ontology directly to author complex argument structures. A more realistic approach is to attempt to create as much annotation as possible in the background without direct user intervention (e.g. using natural language processing techniques to classify arguments automatically).

References

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

[2] D. N. Walton, Fundamentals of Critical Argumentation, Cambridge University Press, New York, USA, 2006.

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

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

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

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

[7] F. H. van Eemeren, R. F. Grootendorst, F. S. Henkemans, Argumentation: Analysis, Evaluation, Presentation, Lawrence Erlbaum Associates, Mahwah NJ, USA, 2002.

[8] T. F. Gordon, H. Prakken, D. Walton, The Carneades model of argument and burden of proof, Artificial Intelligence 171 (10–15) (2007) 875–896.

[9] 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.

[10] C. L. Hamblin, Fallacies, Methuen, London, UK, 1970.

[11] Y. Kalfoglou, M. Schorlemmer, Ontology mapping: the state of the art, Knowledge Engineering Review 18 (1) (2003) 1–31.

[12] 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.

[13] G. Rowe, F. Macagno, C. Reed, D. Walton, Araucaria as a tool for diagramming arguments in teaching and studying philsophy, Teaching Philosophy 29 (2) (2006) 111–124.

[14] C. Mancini, S. Buckingham Shum, Modelling discourse in contested domains: A semiotic and cognitive framework, International Journal of Human Computer Studies 64 (11) (2006) 1154–1171.

[15] S. Buckingham Shum, E. Motta, J. Domingue, ScholOnto: An ontology-based digital library server for research documents and discourse, International Journal of Digital Libraries 3 (3) (2000) 237–248.

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

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

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

[19] 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/`

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

[21] 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, pp. 585–594.

[22] I. Rahwan, F. Zablith, C. Reed, Laying the foundations for a world wide argument web, Artificial Intelligence 171 (10–15) (2007) 897–921.

[23] 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/