Towards Representing and Querying Arguments on the Semantic Web

Iyad Rahwan [a,b,1] P. V. Sakeer [a]
[a] *British University in Dubai, UAE*
[b] *(Fellow) University of Edinburgh, UK*

Abstract. This paper demonstrates the potential of the Semantic Web as a platform for representing, navigating and processing arguments on a global scale. We use the RDF Schema (RDFS) ontology language to specify the ontology of the recently proposed Argument Interchange Format (AIF) and an extension thereof to Toulmin's argument scheme. We build a prototype Web-based system for demonstrating basic querying for argument structures expressed in the Resource Description Framework (RDF). An RDF repository is created using the Sesame open source RDF server, and can be accessed via a user interface that implements various user-defined queries.

Keywords. Argumentation, Semantic Web, Agents, RDF

1. Introduction

Argumentation is 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 a computational or multi-agent system, the *rational judge* could correspond to a particular choice of rules or algorithm for computing the acceptable arguments for deciding the agent that wins the argument. Moreover, the *standpoint* may not necessarily be propositional, and should be taken in the broadest sense (e.g. it may refer to a decision or a value judgement). Finally, the term *controversial* should also be taken in the broad sense to mean "subject to potential conflict."

The theory of argumentation is a rich, interdisciplinary area of research lying across philosophy, communication studies, linguistics, and psychology. Its techniques and results have found a wide range of applications in both theoretical and practical branches of artificial intelligence and computer science [2,3,4].

While argumentation mark-up languages such as those of Araucaria [5], Compendium and ASCE (see [6] for example) already exist, they are primarily a means to enable users to structure arguments through diagrammatic linkage of natural language sentences. Moreover, these mark-up languages do not have rich formal semantics, and are therefore not designed to enable sophisticated automated processing of argumenta-

[1]Correspondence to: Iyad Rahwan, the British University in Dubai, P.O.Box 502216, Dubai, UAE. Tel.: +971 4 367 1959; Fax: +971 4 366 4698; E-mail: irahwan@acm.org.

tive statements. Such semantics may help improve applications of electronic deliberative democracy [7,8,9,10] by enabling citizens to annotate, query and navigate arguments and elements of arguments. Rich formal semantics may also improve capabilities for argumentation among autonomous software agents [11,12,13,14] by enabling the exchange arguments in open multi-agent systems using a standardised format.

In response to the above, an effort towards a standard Argument Interchange Format (AIF) has recently commenced [15]. The aim was to consolidate the work that has already been done in argumentation mark-up languages and multi-agent systems frameworks. It was hoped that this effort will provide a convergence point for theoretical and practical work in this area, and in particular facilitate: (i) argument interchange between agents within a particular multi-agent framework; (ii) argument interchange between agents across separate multi-agent frameworks; (iii) inspection/manipulation of agent arguments through argument visualisation tools; and (iv) interchange between argumentation visualisation tools.

This paper presents a first step towards representing arguments on the World Wide Web using open, rich, and formal semantic annotation. We present building blocks for developing Web-based systems for navigating and querying argument structures expressed in the Resource Description Framework (RDF). The RDF representation of arguments conforms to an ontology of arguments, which is based on the AIF specification and expressed in the RDF Schema language. By expressing the AIF ontology in a standard format (namely RDF), it becomes possible to use a variety of Semantic Web tools (e.g. RDF query engines) to access and process arguments. This approach opens up many possibilities for automatic argument processing on a global scale.

The rest of the paper is organised as follows. In the next Section, we summarise the current state of the Argument Interchange Format specification. In Section 3, we describe how RDF and RDF Schema can be used to specify argument structures. We discuss some related work in Section 4 and conclude the paper in Section 5.

2. The Argument Interchange Format Ontology

In this section, we provide a brief overview of the current state of the Argument Interchange Format. We will use the AIF specification as of April 2006 [15]. The AIF is a core ontology of argument-related concepts. This core ontology is specified in such 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 (di-graph). This di-graph is informally called an *argument network* (AN).

2.1. Nodes

There are two kinds of nodes in the AIF, namely, *information nodes* (I-nodes) and scheme application nodes or *scheme nodes* (S-nodes) for short. Roughly speaking, I-Nodes contain content that represent declarative aspects of the the domain of discourse, such as claims, data, evidence, propositions etc. On the other hand, S-nodes are applications of *schemes*. Such schemes may be considered as domain-independent patterns of reasoning, including but not limited to rules of inference in deductive logics. The present on-

	to *I-node*	to *RA-node*	to *PA-node*
from *I-node*		data/information used in applying an inference	data/information used in applying a preference
from *RA-node*	inferring a conclusion in the form of a claim	inferring a conclusion in the form of a scheme application	inferring a conclusion in the form of a preference application
from *PA-node*	applying preferences among information (goals, beliefs, ..)	applying preferences among inference applications	meta-preferences: applying preferences among preference applications

Table 1. Informal semantics of support.

tology deals with two different types of schemes, namely *inference schemes* and *attack schemes*. Potentially other scheme types could exist, such as evaluation schemes and scenario schemes, which will not be addressed here.

The ontology specifies two types of S-Nodes. If a scheme application node is an application of an inference scheme it is called a *rule of inference application node* (RA-node). If a scheme application node is an application of a preference scheme it is called a *preference application node* (PA-node). Informally, RA-nodes can be seen as applications of rules of inference while PA-nodes can be seen as applications of (possibly abstract) criteria of preference among evaluated nodes.

2.2. Node Attributes

Nodes may possess different attributes that represent things like title, text, creator, type (e.g. decision, action, goal, belief), creation date, evaluation, strength, acceptability, and polarity (e.g. with values of either “pro” or “con”). These attributes may vary and are not part of the core ontology. Attributes may be intrinsic (e.g. “evidence”), or may be derived from other attributes (e.g. “acceptability” of a claim may be based on computing the “strength” of supporting and attacking arguments).

2.3. Edges

According to the AIF core ontology, edges in an argument network can represent all sorts of (directed) relationships between nodes, but do not necessarily have to be labelled with semantic pointers. A node A is said to *support* node B if and only if an edge runs from A to B.[1]

There are two types of edges, namely *scheme edges* and *data edges*. Scheme edges emanate from S-nodes and are meant to support conclusions. These conclusions may either be I-nodes or S-nodes. Data edges emanate from I-nodes, necessarily end in S-nodes, and are meant to supply data, or information, to scheme applications. In this way, one may speak of I-to-S edges (e.g. representing “information,” or “data” supplied to a scheme), S-to-I edges (e.g. representing a “conclusion” supplied by a scheme) and S-to-S edges (e.g. representing one scheme’s attack against another scheme).

[1]Note that this is a rather lose use of the word “support” and is different from the notion of “support between arguments” in which one argument supports the acceptability of another argument.

2.4. Extending the Ontology: Toulmin's Argument Scheme

Philosopher Stephen Toulmin presented a general argument scheme for analysing argumentation. Toulmin's scheme, which has recently become influential in the computational modelling of argumentation, consists of a number of elements which are often depicted graphically as follows:

$$\begin{array}{ccc} & D \longrightarrow & Q, C \\ & | & | \\ \text{since} & W & \text{unless } R \\ & | & \\ & B & \end{array}$$

The various elements are interpreted as follows:

Claim (C): This is the assertion that the argument backs.

Data (D): The evidence (e.g. fact, an example, statistics) that supports the claim.

Warrant (W): This is what holds the argument together, linking the evidence to the claim.

Backing (B): The backing supports the warrant; it acts as an evidence for the warrant.

Rebuttal (R): A rebuttal is an argument that might be made against the claim, and is explicitly acknowledged in the argument.

Qualifier (Q): This element qualifies the conditions under which the argument holds.

An example of an argument expressed according to Toulmin's scheme can be as follows. The war in Irat (a fictional country) is justified (C) because there are weapons of mass destruction (WMDs) in Irat (D) and all countries with weapons of mass destructions must be attacked (W). Countries with WMDs must be attacked because they pose danger to others (B). This argument for war on Irat can be rebutted if the public do not believe the CIA intelligence reports about Irat possessing WMDs (R). Finally, this argument only holds if attacking Irat is less damaging than the potential damage posed by its WMDs (Q).

Toulmin's argument scheme may be represented as an extension of the AIF core ontology. In particular, the concepts of *claim*, *data*, *backing*, *qualifier* and *rebuttal* can all be expressed as sub-classes of I-Node. The concept of *warrant*, on the other hand, is an extension of RA-Nodes. This is because the former concepts all represent passive propositional knowledge, while the warrant is what holds the scheme together. In addition, since I-Nodes cannot be linked directly to one another, we introduce two new extensions of RA-Nodes. The new *qualifier-application* nodes link qualifier nodes to claim nodes, while *rebuttal-application* nodes link rebuttal nodes to claim nodes.

3. Arguments in RDF/RDFS

In this section, we describe the specification of the AIF ontology, and its extension to Toulmin's argument scheme, in RDF Schema.

3.1. Background: XML, RDF and RDFS

The Extensible Mark-up Language (XML) is a W3C standard language for describing document structures by *tagging* parts of documents. XML documents provide means for nesting tagged *elements*, resulting in a directed tree-based structure. The XML Document Type Definition (DTD) and XML Schema languages can be used to describe different *types* of XML documents.

The Resource Description Framework (RDF)[2] is a general framework for describing Internet resources. RDF defines a resource as any object that is uniquely identifiable by an Uniform Resource Identifier (URI). Properties (or attributes) of resources are defined using an object-attribute-value triple, called a *statement*.[3] RDF statements can be represented as 3-tuples, as directed graphs, or using a standard XML-based syntax. The different notations are shown in Figure 1. Attributes are sometimes referred to as *properties* or *predicates*.

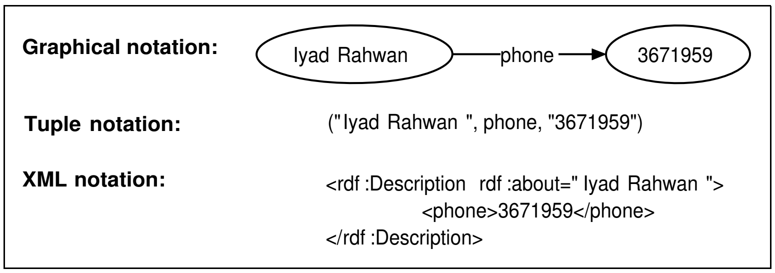

Figure 1. Different notations for RDF statements

Unlike XML, which describes document models in directed-tree-based nesting of elements, RDF's model is based on arbitrary graphs. This structure is better suited for creating conceptual domain models. RDF provides a more concise way of describing rich semantic information about resources. As a result, more efficient representation, querying and processing of domain models become possible.

RDF Schema (RDFS)[4] is an (ontology) language for describing vocabularies in RDF using terms described in the RDF Schema specification. RDFS provides mechanisms for describing characteristics of resources through, for example, domains and ranges of properties, classes of resources, or class taxonomies. RDFS (vocabulary-describing) statements are themselves described using RDF triples.

3.2. AIF and Toulmin's Scheme in RDF Schema

We have first specified the AIF core ontology in RDFS using the Protégé ontology development environment.[5] The main class `Node` was specialised to two types of nodes: `I-Node` and `S-Node`. The `S-Node` class was further specialised to two more classes: `PA-Node` and `RA-Node`. For example, the following RDFS code declares the class `PA-Node` and states that it is a sub-class of the class `S-Node`.

[2] `http://www.w3.org/RDF/`
[3] Sometimes, an *attribute* is referred to as a *property* or a *slot*.
[4] `http://www.w3.org/TR/rdf-schema/`
[5] `http://protege.stanford.edu/`

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

Next, the following elements from Toulmin's scheme were introduced as I-Nodes: claim, data, backing, rebuttal, and qualifier. All these elements represent passive declarative knowledge. Toulmin's warrant was expressed as an RA-Node, since it holds part of the argument together, namely the data nodes and the claim. Similarly, we introduced two other types of RA-Nodes: `Rebuttal-Application` nodes are used to link rebuttal nodes to claims, while `Qualifier-Application` nodes are used to link qualifier nodes to claims. The resulting ontology is represented in Figure 2.

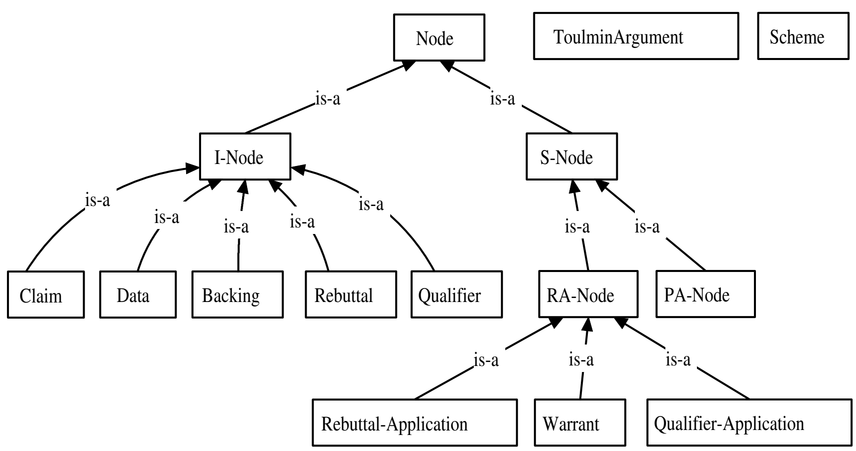

Figure 2. Toulmin argument class hierarchy as an extension of AIF ontology

Note that the concept `ToulminArgument` is a standalone concept. Instances of this concept will represent complete arguments expressed in Toulmin's scheme. Such instances must therefore refer to instances of the various elements of the scheme. The ontology imposes a number of restrictions on these elements and their interrelationships. In particular, each Toulmin argument must contain exactly one claim, exactly one warrant, exactly one qualifier, at least one backing, and at least one datum. As an example, the following RDFS code declares the property `claim` which links instances of `ToulminArgument` to instances of type `Claim`, and states that each `ToulminArgument` must be linked to exactly one `Claim`:

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

In our ontology, we defined various predicates to capture every type of edge, such as those that emanate from backing nodes to warrant nodes, those from warrants to claims, and so on.

Note that according to our ontology, a single claim node can belong to multiple instances of Toulmin arguments, denoting multiple reasons for believing the claim. Similarly, a single data node could contribute to multiple unrelated claims. The RDF graph model enables such flexibility.

With the ontology in place, it is now possible to create instances of the Toulmin argument scheme in RDF. Figure 3 shows the argument mentioned above for justifying the war on Irat. Each box represents an RDF resource, which is an instance of the relevant node type, while edges represent RDF predicates. In addition, all these resources are linked to an instance (named "`IratWar`") of the class `ToulminArgument`, but we omit these links for clarity purposes. In the Figure, we distinguished S-Nodes by dotted boxes although they are not treated differently from the point of view of RDF processing tools.

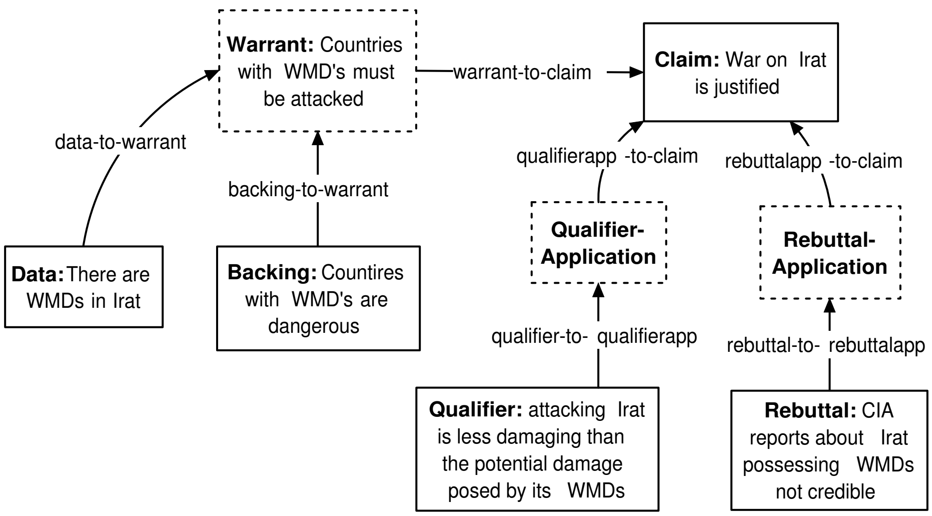

Figure 3. RDF graph for elements of Toulmin argument instance "`IratWar`"

Note that in practice, each of these elements of the argument instance may reside on a different location on the Web. For example, the backing text can be replaced by a reference to a full on-line newspaper article explaining the different dangers countries with WMDs pose. We believe that this feature of RDF could be instrumental for building a layer of argument structures on top of existing Web content.

Finally, we note that the above description is not the only way of representing the Toulmin scheme diagrammatically. Indeed, a Toulmin argument can be represented in more ways than one while, more or less, preserving its semantics. While such representations are outside the scope of this paper, we refer the interested reader to the extensive analysis by Reed and Rowe [16].

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

In order to test our idea, we uploaded the argument instances on Sesame:[6] an open source RDF repository with support for RDF Schema inferencing and querying. Sesame can be deployed on top of a variety of storage systems (relational databases, in-memory, filesystems, keyword indexers, etc.), and offers a large set of tools for developers to leverage the power of RDF and RDF Schema, such as a flexible access API, which supports both local and remote access, and several query languages, such as RQL and SeRQL [17]. Sesame itself was deployed on the Apache Tomcat server, which is essentially a Java servlet container.

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

```
select WARRANT-TEXT, DATA-TEXT, BACKING-TEXT, CLAIM-TEXT
from {WARRANT} kb:scheme-edge-warrant-to-claim {CLAIM},
     {WARRANT} kb:text {WARRANT-TEXT},
     {DATA} kb:data-edge-data-to-warrant {WARRANT},
     {DATA} kb:text {DATA-TEXT},
     {BACKING} kb:data-edge-backing-to-warrant {WARRANT},
     {BACKING} kb:text {BACKING-TEXT},
     {CLAIM} kb:text {CLAIM-TEXT}
where
     CLAIM-TEXT like "War in Irat is justified"
using namespace    kb = http://protege.stanford.edu/kb#
```

The output of the above query returned by Sesame will be the following, showing two arguments. The first justifies war on Irat on the basis of the presence of WMDs. The second argument justifies the war on the basis of removing the country's dictator (a fictional character named "Saddad").

WARRANT-TEXT	DATA-TEXT	BACKING-TEXT	CLAIM-TEXT
Countries with WMD's must be attacked	There are WMD's in Irat	Countries with WMD's are dangerous	War on Irat is justified
Countries ruled by dictators must be attacked	Saddad is a dictator	Dictatorships pose security threats on neighbours	War on Irat is justified

Suppose that after retrieving the first argument, a user or an automated agent is interested in finding out what other claims are supported by the warrant "All Countries with WMD's must be attacked." This information can be found using the following query.

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

```
select WARRANT-TEXT, CLAIM-TEXT
from {WARRANT} kb:scheme-edge-warrant-to-claim {CLAIM},
     {WARRANT} kb:text {WARRANT-TEXT},
     {DATA} kb:data-edge-data-to-warrant {WARRANT},
     {DATA} kb:text {DATA-TEXT},
     {BACKING} kb:data-edge-backing-to-warrant {WARRANT},
     {BACKING} kb:text {BACKING-TEXT},
     {CLAIM} kb:text {CLAIM-TEXT}
where
     WARRANT-TEXT like
                  "All Countries with WMD's must be attacked"
using namespace    kb = http://protege.stanford.edu/kb#
```

The output of this query is as follows:

WARRANT-TEXT	CLAIM-TEXT
Countries with WMD's must be attacked	War on Irat is justified
Countries with WMD's must be attacked	War on USO is justified

In this case, the same warrant used to justify the war against Irat may be used to justify war against the USO (another fictional country).

These queries demonstrate the potential of using the structure of RDF and the expressiveness of RDF query languages to navigate arguments on the Web. Query results can be retrieved via Sesame in XML for further processing. In this way, we could build a more comprehensive system for navigating argument structures through an interactive user interface that triggers such queries.

4. Related Work

A number of argument mark-up languages have been proposed. For example, the Assurance and Safety Case Environment (ASCE)[7] 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* [18].

Another mark-up language was developed for Compendium,[8] 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* [19].

A third mark-up language is the argument-markup language (AML) behind the Araucaria system,[9] an XML-based language [5]. The syntax of AML is specified in a Document Type Definition (DTD) which imposes structural constraints on the form of legal AML documents. AML was primarily produced for use in the Araucaria tool. For example, the DTD could state that the definition of an argument scheme must include a name and any number of critical questions.

[7] http://www.adelard.co.uk/software/asce/
[8] http://www.compendiuminstitute.org/tools/compendium.htm
[9] http://araucaria.computing.dundee.ac.uk/

ClaiMaker and related technologies [20] 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. It provides tools for constructing argument maps, and a server on which they can then be published, navigated, filtered and visualized using the *ClaimFinder* semantic search and navigation tools [21]. This system is based on a specific ontology called the *ScholOnto* ontology [22].

The above attempts at providing argument mark-up languages share some following limitation. Each of these mark-up languages is designed for use with a specific tool, usually for the purpose of facilitating argument visualisation. They were not intended for facilitating inter-operability of arguments among a variety of tools. As a consequence, the semantics of arguments specified using these languages are tightly coupled with particular schemes to be interpreted in a specific tool and according to a specific underlying theory. For example, arguments in Compendium are interpreted in relation to a specific theory of *issue-based information systems*. In order to enable true interoperability of arguments and argument structures, we need an argument description language that can be extended in order to accommodate a variety of argumentation theories and schemes. The AIF, as captured in RDF/RDFS, has the potential to form the basis for such a language.

Another limitation of the above argument mark-up languages is that they are primarily aimed at enabling users to structure arguments through diagramatic linkage of natural language sentences [6]. Hence, these mark-up languages are not designed to process formal logical statements such as those used within multi-agent systems. For example, AML imposes structural limitations on legal arguments, but provides no semantic model. Such semantic model is needed in order to enable the automatic processing of argument structures by software agents.

5. Conclusion

In this paper, we investigated the potential of the Semantic Web as a platform for representing, navigating and processing arguments on a global scale. We used the RDF Schema (RDFS) ontology language to specify the ontology of the recently proposed Argument Interchange Format (AIF) and an extension thereof to Toulmin's argument scheme. We built a prototype Web-based system for demonstrating basic querying for argument structures expressed in the Resource Description Framework (RDF).

Our future plans include extending the AIF core ontology to other argument schemes, such as Walton's schemes for presumptive reasoning [23]. By doing so, we hope to validate the applicability of our approach and identify the limitations of RDF and RDFS for representing argument structures. A more expressive ontology language, such as OWL [24], may be needed.

Another future direction for our work is to build applications that exploit the rich semantics of arguments provided by Semantic Web ontologies. Such applications could range from sophisticated argument processing and navigation tools to support human interaction with argument content, to purely automated applications involving multiple interacting agents operating on Web-based argument structures.

Acknowledgements

The first author is grateful to *AgentLink*[10] and ASPIC [25] for financial support to attend the AgentLink Argumentation Interchange Format Technical Forum in Budapest during September 2005, and to the AIF Technical Forum Group for many stimulating discussions during and after the forum.

References

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

[2] H. Prakken and G. Vreeswijk. Logics for defeasible argumentation. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 4, pages 219–318. Kluwer Academic Publishers, Dordrecht, Netherlands, second edition, 2002.

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

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

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

[6] P. A. Kirschner, S. Buckingham Schum, and C. S. Carr, editors. *Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making*. Springer Verlag, London, 2003.

[7] K. Atkinson, T. Bench-Capon, and P. McBurney. PARMENIDES: facilitating deliberation in democracies. *Artificial Intelligence and Law – T. van Engers and A. Macintosh (editors), Special Issue on eDemocracy*, page (to appear), 2006.

[8] A. Lotov. Internet tools for supporting of lay stakeholders in the framework of the democratic paradigm of environmental decision making. *Journal of Multi-criteria decision analysis*, 12:145–162, 2003.

[9] R. Lüehrs, T. Malsch, and K. Voss. Internet, discourses and democracy. In T. Terano, T. Nishida, A. Namatame, S. Tsumoto, Y. Ohsawa, and T. Washio, editors, *New Frontiers in Artificial Intelligence*, volume 2253 of *Lecture Notes in Computer Science*, pages 67–74. Springer-Verlag, Heidelberg, Germany, 2001.

[10] C. Niculae and S. French. Bringing understanding in societal decision making: Explaining and communicating analyses. *Journal of Multi-criteria decision analysis*, 12:191–202, 2003.

[11] S. Parsons, C. Sierra, and N. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

[12] I. Rahwan. *Interest-based Negotiation in Multi-Agent Systems*. PhD thesis, Department of Information Systems, University of Melbourne, Melbourne, Australia, 2004.

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

[14] F. Sadri, F. Toni, and P. Torroni. Abductive logic programming architecture for negotiating agents. In *Proceedings of the 8th European Conference on Logics in Artificial Intelligence (JELIA-2002)*, volume 2424 of *Lecture Notes in Computer Science*, pages 419–431. Springer Verlag, Berlin, Germany, 2002.

[15] S. Willmott, G. Vreeswijk, C. Chesnevar, M. South, J. McGinis, S. Modgil, I. Rahwan, C. Reed, and G. Simari. Towards an argument interchange format for multiagent systems.

[10] `http://www.agentlink.org`

In Nicolas Maudet, Simon Parsons, and Iyad Rahwan, editors, *Proceedings of the 3rd International Workshop on Argumentation in Multi-Agent Systems (ArgMAS), Hakodate, Japan*, 2006.

[16] C. Reed and G. Rowe. Translating Toulmin diagrams: Theory neutrality in argument representation. *Argumentation*, 19:267–286, 2005.

[17] P. Haase, J. Broekstra, A. Eberhart, and R. Volz. A comparison of rdf query languages. In S. A. McIlraith, D. Plexousakis, and F. van Harmelen, editors, *Proceedings of the 3rd International Semantic Web Conference, Hiroshima, Japan*, volume 3298 of *Lecture Notes in Computer Science*, pages 502–517. Springer Verlag, Heidelberg, Germany, 2004.

[18] L. Emmet and 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*, pages 55–64, New York, USA, 2002. ACM Press.

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

[20] S. Buckingham Shum, V. Uren, G. Li, B. Sereno, and 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 (to appear)*, 2006.

[21] N. Benn, S. Buckingham Shum, and J. Domingue. Integrating scholarly argumentation, texts and community: Towards an ontology and services. In *Proceedings of the Fifth Workshop on Computational Models of Natural Argument (CMNA 2005)*, 2005.

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

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

[24] D. L. McGuinness and F. van Harmelen. Web ontology language (OWL): Overview. Technical report, W3C Working Draft, 31 March 2003.

[25] ASPIC. Argumentation service platform with integrated components, a European Commission-funded research project (no. ist-fp6-002307), 2004.