AN ARGUMENTATION CORE ONTOLOGY AS THE CENTERPIECE OF A MYRIAD OF ARGUMENTATION FORMATS

Bart Verheij
Artificial Intelligence, University of Groningen
b.verheij@ai.rug.nl, http://www.ai.rug.nl/~verheij/

My starting point in this position statement is that it is too ambitious to aim for the development of a single comprehensive argumentation format, for the simple reason that there are many argumentation theories around (formal and informal), all with their own sets of concepts and points of focus. In part, this is the nature of things: specific purposes and contexts ask for specific variants of argumentation theory, and hence for specific argumentation formats. Moreover, argumentation is such a complex, multidisciplinary topic with such a multifaceted history that a single overarching, comprehensive format seems to be out of reach.

I also claim that it is not necessary to develop a single argumentation format for interchange purposes. I propose to focus initially on a related, but less bold aim: the development of an argumentation core ontology. Such an ontology should be minimal in the sense that it should only express what can be generally agreed upon. The argumentation core ontology will contain concepts, such as argument, reason, statement, dialogue, dialogue participant, etc. and relations, such as having a conclusion (between an argument and a statement), being a reason for (between statements) and being committed to (between a dialogue participant and a statement).

A common core of concepts and conceptual relations of argumentation like an argumentation core ontology need be neither comprehensive nor detailed. For instance, the precise relation between reasons and arguments has found different explications in the literature and hence may not be a part of a core ontology. However the idea that arguments contain reasons is assumedly generally agreed upon and is hence a candidate conceptual relation in an argumentation core ontology.

Does this view impede the ideal of an argumentation interchange format? Yes and no. Yes, since a core ontology - or any ontology for that matter - is not in itself an interchange format. No, since an ontology can be the centerpiece of a myriad of special-purpose argumentation formats (cf. Figure 1). Each argumentation format should use the argumentation core ontology as its starting point and provide a translation back into the core ontology. In this situation, translations between argumentation formats are optional (hence the dotted arrows in the figure) and can be developed whenever considered useful. The core ontology provides the glue.

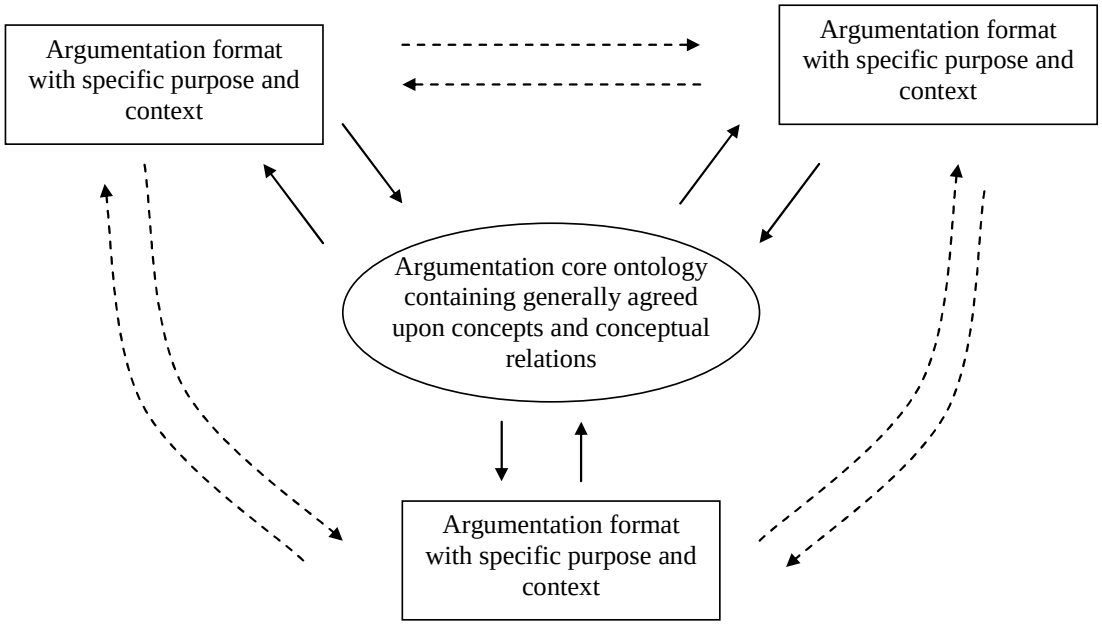

Figure 1: An argumentation core ontology as the centerpiece of argumentation formats

Note that in this view, argumentation formats are conceived of as special-purpose, detailed and temporary, whereas the argumentation ontology is general, provides a bird's eye view and is long-lasting. Whereas the core ontology provides the basic concepts and relations in the topic of argumentation, argumentation formats are used for the storing, exchanging and processing of argumentative information.

What technology should be used for the development of an argumentation core ontology and for argumentation formats? In my opinion, an argumentation core ontology should be developed using today's *de facto* research standard for web-oriented ontologies: the Web Ontology Language OWL (http://www.w3.org/2004/OWL/) in combination with the open-source ontology editor Protégé (http://protege.stanford.edu/). I think that OWL is to be preferred over RDF (or more accurately: RDF Schema). Although RDF is closer to being today's industry standard, OWL's extended expressiveness and research appeal make it preferable.

Argumentation formats should be developed in XML Schema, the preferred schema language associated with XML. Documents containing argumentative information can them be structured as XML documents following a particular XML Schema argumentation format. XML's down-to-earth syntax (which is also readable for humans) and data model and the wealth of available tools make XML the best option. The transformation language XSLT provides extensive and straightforward possibilities for translating between XML documents, and hence for the translation from one argumentation format to another. XSLT can also be used to map an argumentation format onto an argumentation ontology.

What would an argumentation ontology in OWL/Protégé look like? Figure 2 provides an example of a class hierarchy and a list of properties that could be part of an argumentation ontology. There are for instance classes for arguments, reasons and

statements. Reasons have pro-reasons and con-reasons as subclasses. There are properties indicating for instance conflicts, antecedents and opposites.

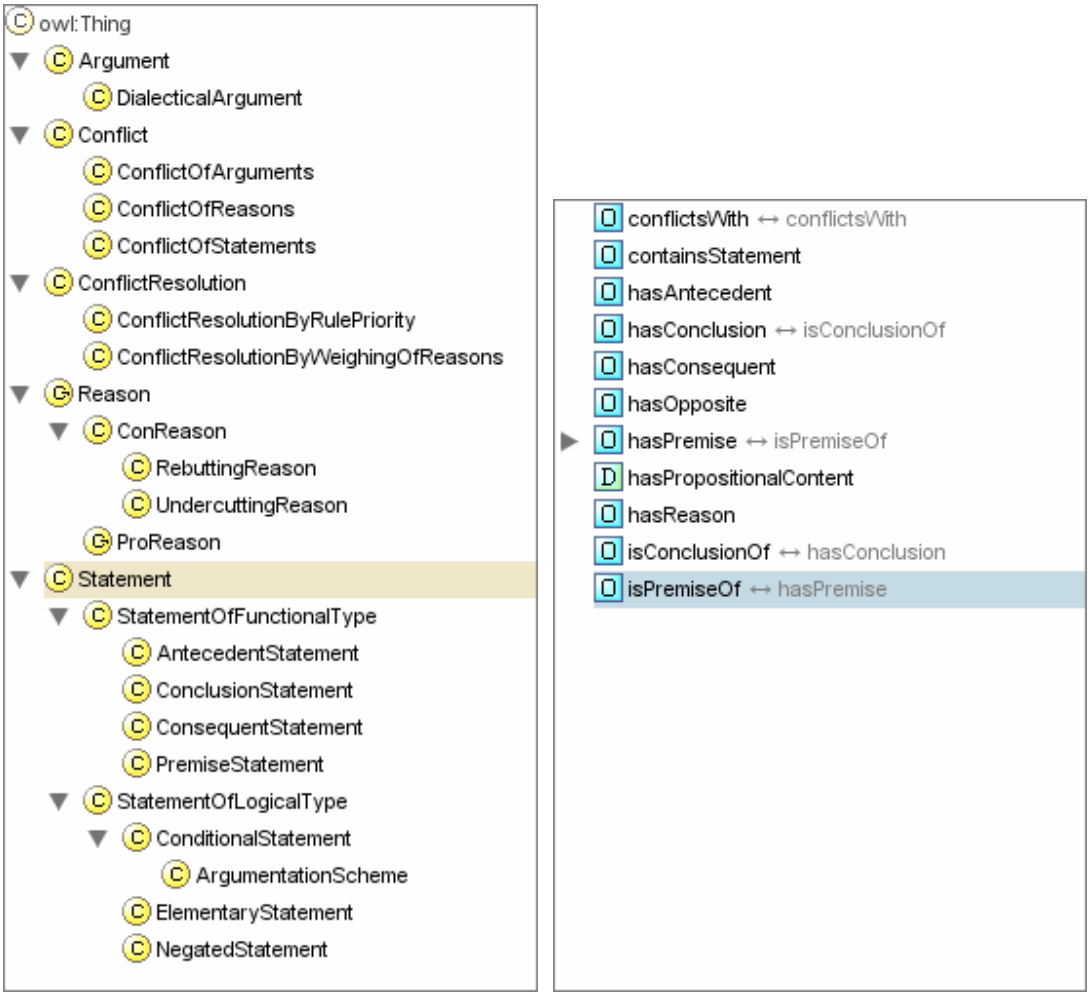

Figure 2: An example of a class hierarchy and of a list of properties in an argumentation ontology (made using Protégé's OWL plugin)

And what would be an example of an argumentation format in XML? Figure 3 shows the graphical representation of an argumentation format specified using XML Schema. According to this format, a theory consists of two parts: a sequence of assumptions and a sequence of issues. Both consist of sentences, which can be of three types: elementary sentences, conditional sentences and negated sentences.

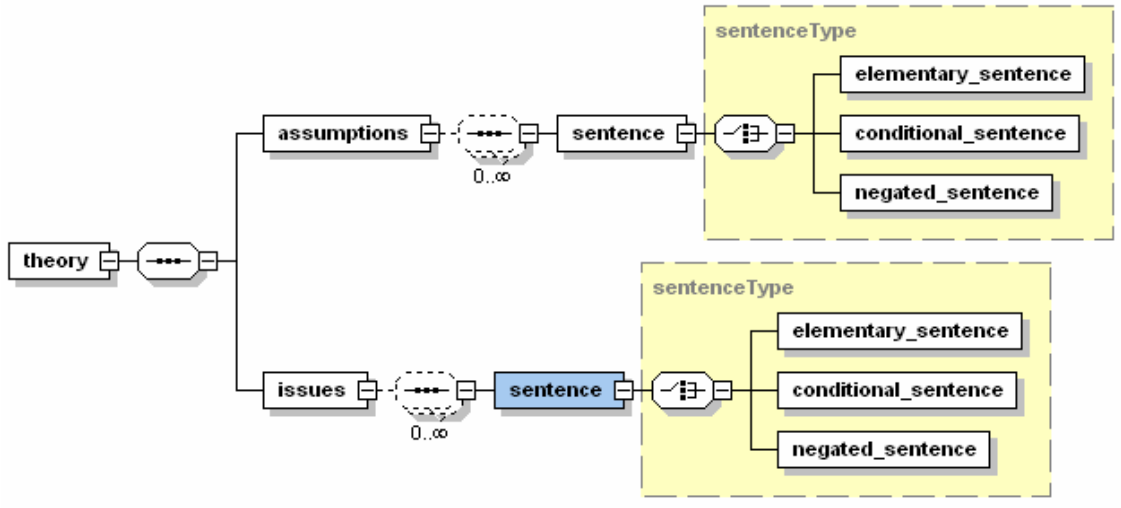

Figure 3: A example of a special purpose argumentation format
(made using Altova's XMLSpy)

Figure 4 shows an instance of a document corresponding to the argumentation format in Figure 3. Here there are two assumptions: the elementary sentences 'John is a thief' and 'Thieves are punishable'. The sentence 'John is punishable' is an issue.

```
<?xml version="1.0" encoding="UTF-8"?>
<theory xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="aif.xsd">
   <assumptions>
      <sentence>
         <elementary_sentence>John is a thief</elementary_sentence>
      </sentence>
      <sentence>
         <elementary_sentence>Thieves are punishable</elementary_sentence>
      </sentence>
   </assumptions>
   <issues>
      <sentence>
         <elementary_sentence>John is punishable</elementary_sentence>
      </sentence>
   </issues>
</theory>
```

Figure 4: An example of a document containing argumentative information
corresponding to a special purpose argumentation format

In an ideal world, the situation would be somewhat more elaborate than in Figure 1: each argumentation format should go with its own special-purpose ontology, which on its turn is coupled with the argumentation core ontology (Figure 5).

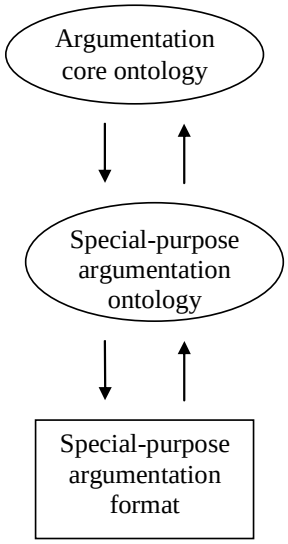

Figure 5: An argumentation core ontology, a special-purpose argumentation format and a special-purpose argumentation format

Summary of recommendations

- Develop a broad argumentation core ontology in OWL using Protégé, in which only generally agreed upon concepts and conceptual relations are specified.
- Develop special purpose argumentation formats using XML technology (in particular XML Schema for argumentation formats and XSLT for translations between formats and onto ontologies).
- Base each special purpose argumentation format on the argumentation core ontology and provide a translation back into the core ontology.
- Associate special purpose argumentation formats with special purpose ontologies based on the argumentation core ontology.

Appendix 1: OWL-file corresponding to the ontology in Figure 2

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
   xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns="http://www.owl-ontologies.com/unnamed.owl#"
 xml:base="http://www.owl-ontologies.com/unnamed.owl">
 <owl:Ontology rdf:about="">
   <owl:imports
          rdf:resource="http://protege.stanford.edu/plugins/owl/protege"/>
 </owl:Ontology>
 <owl:Class rdf:ID="ConReason">
   <owl:disjointWith>
     <owl:Class rdf:ID="ProReason"/>
   </owl:disjointWith>
```

```
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Reason"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="ArgumentationScheme">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="ConditionalStatement"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="RebuttingReason">
    <rdfs:subClassOf rdf:resource="#ConReason"/>
  </owl:Class>
  <owl:Class rdf:ID="PremiseStatement">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="StatementOfFunctionalType"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="ConflictResolutionByRulePriority">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="ConflictResolution"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="ConflictOfStatements">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Conflict"/>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:minCardinality
           rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >2</owl:minCardinality>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="containsStatement"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#ConditionalStatement">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="StatementOfLogicalType"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#ProReason">
    <owl:disjointWith rdf:resource="#ConReason"/>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Reason"/>
    </rdfs:subClassOf>
    <protege:subclassesDisjoint
           rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >true</protege:subclassesDisjoint>
  </owl:Class>
  <owl:Class rdf:ID="AntecedentStatement">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#StatementOfFunctionalType"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#StatementOfFunctionalType">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Statement"/>
```

```
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="NegatedStatement">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#StatementOfLogicalType"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Argument"/>
  <owl:Class rdf:ID="ConclusionStatement">
    <rdfs:subClassOf rdf:resource="#StatementOfFunctionalType"/>
  </owl:Class>
  <owl:Class rdf:about="#Reason">
    <protege:subclassesDisjoint
           rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >true</protege:subclassesDisjoint>
  </owl:Class>
  <owl:Class rdf:ID="UndercuttingReason">
    <rdfs:subClassOf rdf:resource="#ConReason"/>
  </owl:Class>
  <owl:Class rdf:ID="ConflictOfArguments">
    <rdfs:subClassOf rdf:resource="#Conflict"/>
  </owl:Class>
  <owl:Class rdf:ID="ConsequentStatement">
    <rdfs:subClassOf rdf:resource="#StatementOfFunctionalType"/>
  </owl:Class>
  <owl:Class rdf:about="#StatementOfLogicalType">
    <rdfs:subClassOf rdf:resource="#Statement"/>
  </owl:Class>
  <owl:Class rdf:ID="ElementaryStatement">
    <rdfs:subClassOf rdf:resource="#StatementOfLogicalType"/>
  </owl:Class>
  <owl:Class rdf:ID="ConflictResolutionByWeighingOfReasons">
    <rdfs:subClassOf rdf:resource="#ConflictResolution"/>
  </owl:Class>
  <owl:Class rdf:ID="ConflictOfReasons">
    <rdfs:subClassOf rdf:resource="#Conflict"/>
  </owl:Class>
  <owl:Class rdf:ID="DialecticalArgument">
    <rdfs:subClassOf rdf:resource="#Argument"/>
  </owl:Class>
  <owl:ObjectProperty rdf:ID="hasAntecedent">
    <rdfs:domain rdf:resource="#ConditionalStatement"/>
    <rdfs:range rdf:resource="#Statement"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="isPremiseOf">
    <rdfs:range rdf:resource="#Argument"/>
    <rdfs:domain rdf:resource="#Statement"/>
    <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="hasPremise"/>
    </owl:inverseOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="conflictsWith">
    <rdfs:range rdf:resource="#Statement"/>
    <owl:inverseOf rdf:resource="#conflictsWith"/>
    <rdf:type
           rdf:resource="http://www.w3.org/2002/07/owl#SymmetricProperty"/>
    <rdfs:domain rdf:resource="#Statement"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#hasPremise">
```

```
    <rdfs:range rdf:resource="#Statement"/>
    <owl:inverseOf rdf:resource="#isPremiseOf"/>
    <rdfs:domain rdf:resource="#Argument"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#containsStatement">
    <rdfs:domain rdf:resource="#ConflictOfStatements"/>
    <rdfs:range rdf:resource="#Statement"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasOpposite">
    <rdfs:range rdf:resource="#Statement"/>
    <rdfs:domain rdf:resource="#NegatedStatement"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasReason">
    <rdfs:range rdf:resource="#Reason"/>
    <rdfs:domain rdf:resource="#Argument"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasConclusion">
    <rdf:type
           rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <rdfs:domain rdf:resource="#Argument"/>
    <owl:inverseOf>
      <owl:InverseFunctionalProperty rdf:ID="isConclusionOf"/>
    </owl:inverseOf>
    <rdfs:range rdf:resource="#ConclusionStatement"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasConsequent">
    <rdfs:range rdf:resource="#Statement"/>
    <rdfs:domain rdf:resource="#ConditionalStatement"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasConPremise">
    <rdfs:subPropertyOf rdf:resource="#hasPremise"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasProPremise">
    <rdfs:subPropertyOf rdf:resource="#hasPremise"/>
  </owl:ObjectProperty>
  <owl:FunctionalProperty rdf:ID="hasPropositionalContent">
    <rdfs:domain rdf:resource="#Statement"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdf:type
           rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  </owl:FunctionalProperty>
  <owl:InverseFunctionalProperty rdf:about="#isConclusionOf">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <rdfs:range rdf:resource="#Argument"/>
    <rdfs:domain rdf:resource="#ConclusionStatement"/>
    <owl:inverseOf rdf:resource="#hasConclusion"/>
  </owl:InverseFunctionalProperty>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 1.3, Build 225.4)
          http://protege.stanford.edu -->
```

Appendix 2: the XML Schema of Figure 3

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
   <xs:complexType name="sentenceType">
```

```
      <xs:choice>
        <xs:element name="elementary_sentence"/>
        <xs:element name="conditional_sentence"/>
        <xs:element name="negated_sentence"/>
      </xs:choice>
    </xs:complexType>
    <xs:element name="theory">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="assumptions">
            <xs:complexType>
              <xs:sequence minOccurs="0" maxOccurs="unbounded">
                <xs:element name="sentence" type="sentenceType"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="issues">
            <xs:complexType>
              <xs:sequence minOccurs="0" maxOccurs="unbounded">
                <xs:element name="sentence" type="sentenceType"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
</xs:schema>
```