Semantic Web: Schism of the Languages

Michael Kifer

State University of New York
at Stony Brook
USA
Where is Stony Brook?
Outline

- What is the Semantic Web?
- Ontologies: Conceptual models of the Semantic Web
- Semantic Web languages
  - Description Logic
  - Rule-based languages
  - The nature of the schism
  - Rule Interchange Format (RIF)
- Conclusions
  - Problems
  - Outlook
What is Semantic Web?

- Electricity of the 21st Century:
  Machine-understandable information everywhere

- Semantically meaningful search
  - Suggest fine restaurants according to X taste within walking distance from Y

- Semantic Web services: Software agents doing useful stuff for us autonomously
  - Organize my trip to X, Y, Z, back to W for 3 days, and leave one morning for a meeting in V. Need to be in T by date D. Take care of the tickets, hotels, transportation, visas, etc.
The Silver Bullet of the Semantic Web

- Ontologies
  - A database schema (like E-R)
    - Classification hierarchies
    - Typing
    - Constraints
  - ... and some more
    - concept `human` is same-as concept `person`
    - property `prerequisite` is transitive
    - property `spouse` is symmetric
The definitional part

- Dean = HeadOf of College
- GradStudent = Student with Status ∈ \{G1,…,G5\}
- HasHead = inverse of HeadOf
- ...

Why Ontologies?

- They give precise meaning to terms (concepts, properties) in an application domain.

  => applications can talk to each other
  - can unambiguously interpret the info they exchange
  - even if they use different ontologies

  => semantic interoperability
Beyond Ontologies

- Need more than just unary (classes) and binary (properties) predicates
  - E-R diagrams go further than ontologies in this respect
- Need machinery for defining more complex concepts & properties
- Need more than just concepts and properties
  - Processes (e.g., for Web services)
  - Authorization schemes (can be very complex)
  - Networks of trust

```
PreferredSupplier(?Supplier, ?Product)
if ?Supplier Supplies ?Product to another project within same organization and …
```
How are Ontologies Represented?

- **Description logic (DL)**
  - Subset of classical logic (dates back to late 70’s: *KL-1*, Brachman et. al.)
  - W3C standard, *OWL-DL* – Web Ontology Language, is based on a description logic called *SHOIN(D)*
- **Pros:**
  - Predictable complexity
  - Sometimes can do surprisingly non-trivial reasoning
- **Cons:**
  - Can be awkward to use (strange, highly controlled language)
  - Quite limited
    - => Declarative ontologies are processed with Java
  - Little experience with real life problems
How are Ontologies Represented?

- **Rule-based languages**
  - **Pros:**
    - Typically much easier to use than DL, more expressive (where it counts for a man-on-the-street)
    - Efficient implementations
    - Vast experience with real life problems
    - Complete applications can be built declaratively
  - **Cons:**
    - No standard, variety of semantics
      - W3C has established a Rule Interchange Format working group
    - Has limitations in expressivity compared to DL in some areas
Overview of Description Logic

- **Primitive Concepts** – unary predicates; define objects that populate concepts
  - E.g., *Student, Employee*

- **Properties of objects** – binary predicates:
  - E.g., *HasName(john, ‘John Doe’)*

- **Objects** – constants:
  - E.g., *John, Mary*

- **Terminological descriptions**: *T-Box*
  - Expressions for constructing new concepts:
    - E.g., *graduate-students-who-took-2-or-more-courses*
  - Assertions about relationships between concepts
    - E.g., *spouse is a symmetric relationship*
Overview of Description Logic (cont’d)

- **World descriptions**: A-Box
  - Assertions about individuals being members of classes (concepts) or having properties
    - E.g., Mary ∈ Student, John hasAge 44

- **Inference**
  - Whether a concept definition is satisfiable
  - Whether some concept is a sub-concept of another
  - Whether two concepts are disjoint

- In a sufficiently rich DL, the above three are equivalent
## DL Terminological Descriptions – Constructors

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A</strong> <code>(Student)</code></td>
<td>Name of a concept</td>
</tr>
<tr>
<td><strong>T, ⊥</strong></td>
<td>Anything, Nothing</td>
</tr>
<tr>
<td>¬C</td>
<td>Not C (where C is a concept expression)</td>
</tr>
<tr>
<td>¬Student</td>
<td>Non-students</td>
</tr>
<tr>
<td>C ∩ D, C ∪ D</td>
<td>C intersect D, C union D</td>
</tr>
<tr>
<td>Student ∩ Employee</td>
<td>Students who are employees</td>
</tr>
<tr>
<td>∀ <code>Prop.Concept</code></td>
<td>Concept defined as `{x</td>
</tr>
<tr>
<td>∀ <code>Color.BrightColor</code></td>
<td>Brightly-colored things</td>
</tr>
<tr>
<td>∃ <code>Prop.Concept</code></td>
<td>Concept defined as `{x</td>
</tr>
<tr>
<td>≤_n <code>Prop.Concept</code></td>
<td>`{x</td>
</tr>
<tr>
<td>≤_3 <code>Publication.GoodPaper</code></td>
<td>People who wrote less than 4 good papers</td>
</tr>
<tr>
<td>≥_n <code>Prop.Concept</code></td>
<td>`{x</td>
</tr>
<tr>
<td><code>Prop⁻, Prop⁺</code></td>
<td>Inverse of Prop, transitive closure of Prop</td>
</tr>
</tbody>
</table>
### DL Terminological Assertions

<table>
<thead>
<tr>
<th>Assertion</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C \subseteq D )</td>
<td>All objects of concept ( C ) are also in ( D )</td>
</tr>
<tr>
<td>Student ( \subseteq (\exists \text{ Takes}.\text{Class}) )</td>
<td>All student take classes</td>
</tr>
<tr>
<td>Prop1 ( \subseteq ) Prop2</td>
<td>Prop1 is a subproperty (subrelation) of Prop2</td>
</tr>
<tr>
<td>hasSon ( \subseteq ) hasChild</td>
<td>one who has sons has children as well</td>
</tr>
<tr>
<td>( C = D )</td>
<td>( C ) and ( D ) are the same property (( C \subseteq D ) &amp; ( D \subseteq C )), where ( C ) is a concept name. Usually used in definitions of new concepts</td>
</tr>
<tr>
<td>Prop1 = Prop2</td>
<td>Prop1 ( \subseteq ) Prop2 and Prop2 ( \subseteq ) Prop1. Usually used for definitions of new properties</td>
</tr>
<tr>
<td>Prop in Roles_+</td>
<td>Prop is transitive</td>
</tr>
</tbody>
</table>
## World Descriptions (A-Box)

<table>
<thead>
<tr>
<th>College(engineering)</th>
<th>engineering is a kind of a college in a university</th>
</tr>
</thead>
<tbody>
<tr>
<td>headOf(bob, engineering)</td>
<td>bob manages the engineering college</td>
</tr>
<tr>
<td>hasHead(engineering, bob)</td>
<td>engineering college has bob as manager</td>
</tr>
</tbody>
</table>
Ontological Knowledge in DL

- **Concept of a Dean:**
  \[ \text{Dean} = \exists \text{Manages}.\text{College} \]

- **Mother:**
  \[ \text{Mother} = \text{Woman} \sqcap \exists \text{HasChild}.\text{Person} \]

- **Student is a subclass of Person:**
  \[ \text{Student} \subseteq \text{Person} \]
“Surprises” with DLs

- Can be awkward
  - Concept of “Grandmother”
    \[
    \text{Grandmother} = \text{Person} \cap \text{Female} \cap \exists \text{hasChild} . (\exists \text{hasChild}.\text{Person})
    \]

- Insufficiently expressive
  - Concept of “Uncle” (cannot be expressed in DL)
    \[
    \forall X,Y,Z \ (\text{Uncle}(X,Y) \iff \text{Brother}(X,Z) \& \text{Parent}(Z,Y))
    \]
DL Reasoning

- Can answer hard questions
  - Terminological \((potentially\) useful for integration of information, schema checking, query optimization)
    - Whether \(\text{ConceptExpr}_1 \subseteq \text{ConceptExpr}_2\)
    - Whether \(\text{ConceptExpr}\) is satisfiable
      - \(N\text{ExpTime}\) for OWL-DL (the most popular DL)
  - Query answering (most common use)
    - Whether a concrete individual \(x\) belongs to \(\text{ConceptExpr}\)
    - Find all \(x\) such that \(\text{ConceptExpr}(x)\) is entailed by the ontology

- Query answering not really scalable
OWL (Web Ontology Language)

- **OWL-DL**
  - The most-used dialect of OWL
  - Includes more or less the above mentioned DL constructs with restrictions (e.g., no cardinality constraints on transitive properties)

- **OWL-Full**
  - Attempt to make OWL compatible with RDFS (Resource Description Framework Schema), an earlier *ill-conceived* standard
  - Terminological reasoning undecidable
  - Not used that much

- **OWL-Lite**
  - A simpler subset of OWL-DL
  - Rarely used
Rule Languages

- Ontologies are represented as facts + rules
  - Terminological descriptions
    - subclass(Student, Person)
    - $\forall ?M?C \ Mother(?M,?C) \leftarrow \ Woman(?M) \& \ hasChild(?M,?C) \& \ Person(?C)$
    - $\forall ?U?P?F \ Uncle(?U,?P) \leftarrow \ Father(?F,?P) \& \ Brother(?U,?F)$
    - Types:
      - Mother(Woman,Person)
      - Father(Man,Person)

- World descriptions
  - father(John,Bob)
  - hasChild(Mary,Bob)
Rule-based Reasoning

- **Terminological**
  - Whether Concept\(_1\) \(\subseteq\) Concept\(_2\)
  - Whether Concept is satisfiable
    - Both problems are undecidable in general
    - But Concept\(_1\) \(\subseteq\) Concept\(_2\) or nonempty(Concept)
      
      *in a particular world* is polynomial time in the size of the data

- **Query answering**
  - Whether \(\chi\) \(\in\) Concept, \(\chi\)-individual
  - Find all \(\chi\) such that \(\chi\) \(\in\) Concept is entailed by the ontology
    - Both polynomial (in the size of the data), very scalable
Weaknesses of Rule Systems

- **Existential information**
  
  Every person has a father
  
  \[ \forall ?P \exists ?F \text{ Father}(?P, ?F) \leftarrow \text{ Person}(?P) \]
  
  cannot be expressed directly in a rule-based language

- **For most purposes existentials can be approximated with Skolem functions**

  \[ \forall ?P \text{ Father}(?P, _#(?F)) \leftarrow \text{ Person}(?P) \]

  `_#` is a unique new function symbol

- **Even harder to represent disjunctive information**

  Example: \text{ Dead}(\text{John}) \text{ or Alive}(\text{John})
The Nature of the Schism

- Rule-based and DL languages may appear very close – even syntactically.
- But they are very different semantically:
  - DL is a subset of classical logic.
  - Rule-based languages have certain capabilities of the second-order logic:
    - despite deceptively first-order-looking syntax!
The Schism: A Travel Example

take-a-flight(?From,?To) ← flight(?From,?To)
take-a-train(?From,?To) ← not flight(?From,?To)
flight(London,Edinburgh)
flight(London,Amsterdam)

Query: take-a-train(London,Bristol) ?

In classical logic:
- can only derive take-a-train(London,Bristol) or flight(London,Bristol)
  - Cannot derive take-a-train(London,Bristol)

In rule-based logics:
- Cannot derive flight(London,Bristol) ?
  - conclude not flight(London,Bristol)
  - derive take-a-train(London,Bristol)
The Logic of Rules

- The kind of non-classical reasoning used by rule languages is variously known as
  - Default Negation
  - Negation as failure (NAF)
  - Closed-world assumption (CWA)
  - Common-sense reasoning
  - Nonmonotonic logic
- Precise formalization is not exactly trivial, and there is more than one
The Logic of Rules (cont’d)

- Can CWA be simulated in classical logic?
  
  \[
  \text{take-a-flight(?From,?To) } \iff \text{ flight(?From,?To)}
  \]
  
  \[
  \text{take-a-train(?From,?To) } \iff \neg \text{ flight(?From,?To)}
  \]
  
  \[
  \text{flight(London,Edinburgh)}
  \]
  
  \[
  \text{flight(London,Amsterdam)}
  \]

- now can derive

  \[
  \neg \text{flight(London,Bristol)}
  \]
  
  \[
  \text{take-a-train(London,Bristol)}
  \]

  … as if using CWA

- This can be done in some cases but \textit{not in general}
Transitive closure

\[
\text{trip}(\text{?From}, \text{?To}) \leftarrow \text{flight}(\text{?From}, \text{?To})
\]

\[
\text{trip}(\text{?From}, \text{?To}) \leftarrow \text{trip}(\text{?From}, \text{?Mid}) \text{ and } \text{trip}(\text{?Mid}, \text{?To})
\]

In classical logic:

\[
\text{trip} \text{ is transitive & contains } \text{flight}
\]

In rule languages:

\[
\text{trip} \equiv \text{transitive closure of } \text{flight}
\]

Transitive closure \textit{cannot} be expressed in classical logic

The semantics of rule languages is based on minimal models and preference relations among them
Bridging the Schism

- ... and a war was ranging
- Proposed ways out
  - MKNF – Autoepistemic logic of Minimal Belief and Negation as Failure (Lifschitz, 1991)
    - Motik & Rosati 2006
  - Oracle-based solutions – treat DL-based ontologies as black boxes
    - Eiter et. al. 2003/4
  - Autoepistemic Logic
    - De Bruijn et al., 2006
But Schism Not Going Away

- Combining rules and ontologies for what?
- What is the role of rules and of DL reasoning in the grand schema of things?
  - Does DL’s A-box reasoning make sense?
  - DL’s for schema reasoning – rules for querying data?
  - Dual use of DL’s T-box (Motik):
    - Classical for schema reasoning
    - Nonclassical (together with rules) for querying
- Danger of over-selling wrong technologies for wrong tasks
Enter the Rule Interchange Format (RIF)

- Rules landscape is fragmented:
  - Too many kinds of rules
  - Too many syntaxes
  - Too many semantics
  - Too many commercial interests
  - Most industrial uses of rules are not based on sound logical theories
- Only hope is to be able to exchange rule sets “of the same kind” through a common well-defined language (RIF)
RIF Highlights

- Well-defined syntax and semantics
- RIF Core
- RIF Dialects extending the core
  - Rule sets to be exchanged through dialects in semantics-preserving ways
- Some may choose to use RIF as an actual language and not just an exchange medium
Schism May Widen

- OWL may add some rules
  - Will solve the “uncle” problem, but loose decidability

- RIF will have loosely coupled interface to OWL
  - Probably Eiter et. al.-like
  - People will figure out how to use both
  - RDF will be integrated more tightly – through F-logic like frames
How Will RIF Work?

Rule Set 2

Some RIF dialect

Rule Set 1
- Each dialect must have precise syntax & semantics
- Dialects can extend each other
- All must extend the core (both syntactically and semantically)
RIF Core

- Basically Horn clauses with extensions
  - Most notably: frames a la F-logic
- A Web language
  - URIs as constants, concepts, etc.
  - Support for XML data types
- First draft (WD1) – released end of April 2007
  - Boley & Kifer – technical editors
- Other drafts: Use Cases and Requirements – published earlier
Planned RIF Dialects

- Committed to eventually support roughly the following kinds of dialects:
  - PR: production rules (interest from the industry)
  - ECA rules (trigger-like)
  - LP: a logic programming-based dialect
  - FO: first-order logic (some kind of)
  - Constraints
  - HiLog like higher-order extensions
Research Issues

- Practical unification of knowledge representation with DL and Rules – still very much an open problem
- Uncertain and inconsistent information coming from different sources
- Algorithms and approaches for scalable inference on the Web
  - Massive amounts of data and rules
- Making logic accessible to domain experts (who are not logicians)
  - High-level logic languages, like F-logic, etc.
  - Visual editors for creating ontologies and rules
  - Generation of explanations for inference
- Finding a killer application!
Research Issues (cont’d)

- Information integration
  - Query subsumption:
    - whether $\text{Ontology } \models \text{Query}_1 \implies \text{Query}_2$ for a given Ontology
    - or whether $\text{Ontology } \models \text{Query}_1 \implies \text{Query}_2$, $\forall \text{Ontology}$
  - These problems are undecidable in general; many special cases have high complexity
  - Challenge is to identify useful decidable classes of ontologies/queries
  - Useful classes with low complexity
- Modeling exceedingly complex application domains like Web services (see next)
What is a Semantic Web Service?

- A good candidate for the “killer application” that will make the semantic Web a must-have
- Research issues in Semantic Web Services
  - Advertising and discovery
    - Logical service descriptions will enable agents (people, other services) to discover services that can potentially fulfill given needs
  - Contracting
    - Logic-based descriptions of the legal obligations of parties. Will potentially require legislation to enforce such contracts
  - Process modeling & enactment
    - Will enable automatic composition, invocation, and monitoring of services
Outlook

- OWL - 1st step towards the Semantic Web
- Clear that rules are coming in the form of RIF
- Must understand how these two *should* work together in practice (not just how they *could* work)
- Cause for optimism:
  Strong industrial interest & participation in virtually all aspects of the Semantic Web
  - OWL
  - RIF
  - Semantic Web Services
Questions?