Visualizing and Specifying Ontologies using Diagrammatic Logics

Who? Ian Oliver¹, John Howse², Gem Stapleton², Esko Nuutila³ and Seppo Törmä³

From? Nokia Research¹, University of Brighton² and Helsinki University of Technology³
ian.oliver@nokia.com
{john.howse,g.e.stapleton}@brighton.ac.uk
{esko.nuutila,seppo.torma}@hut.fi

Support ¹³Supported by the TEKES ICT SHOK DIEM Project. ²Supported by EPSRC grants EP/E011160/1 for the Visualization with Euler Diagrams Project

When? December 1, 2009
Contribution of the Paper

Proposes a diagrammatic framework for:

1. specifying the concepts and roles
2. placing constraints on the concepts and roles
3. depicting instances
4. defining queries
5. reasoning about the ontology
Motivation

Tasks

1. semantic understanding $\rightarrow$ syntactic specification in chosen notation
2. syntactic specification $\rightarrow$ semantic understanding
3. reason about the specification to further understand its logical consequences.

Can be broken into subtasks and more tasks added.

Question

How can we help with the tasks?

Answer

Make the syntax more accessible to the user

We propose that some users may benefit from visual/diagrammatic syntax
Motivation

Why diagrams?

1. some people find symbolic notations off-putting
2. syntax can be well-matched to semantics
3. can be formalized

*a major alternative for increasing the usability of description logics as a modelling language... is to implement interfaces where the user can specify the representation structures through graphical operations* [Nardi and Brachman, DL Handbook].

Reasoning too, not just representation.
Designing Diagrammatic Notations

Question
What makes a good diagram (notation)?
1. well matched to its semantics [Gurr]
2. presence of free rides [Shimojima]
3. designed for the intended task [Grawemeyer]
4. sufficiently expressive
Well Matched to Semantics

Set of elements in set $A$ well matched

Set of elements not in set $A$ NOT well matched
Well Matched to Semantics

- Set of elements in set $A$ well matched
- Set of elements not in set $A$ NOT well matched
- $B$ is a subset of $A$ well matched
- $B$ is disjoint from $A$ NOT well matched

under the respective semantics
Free Rides

Set of elements in set $A$ well matched

Set of elements not in set $A$ NOT well matched

under the respective semantics

$B$ is a subset of $A$ well matched

$B$ is disjoint from $A$ NOT well matched

$B$ is a subset of $A$
$C$ is a subset of $B$

FOR FREE: $C$ is a subset of $A$ FOR FREE: $C$ is disjoint from $A$
Case Study: Nokia Meetings

Concepts

1. Meeting: this represents the notion of a meeting,
2. Location: every meeting will have a location (in this ontology, exactly one location),
3. Topic: every meeting will have at least one topic,
4. AgendalItem: every meeting will at least one item on its agenda,
5. Participant: every meeting will have at least one host and a set of participants (including the host),
6. Document: each agenda item will have a set of documents (possibly none), and be the responsibility of exactly one participant,
7. Name: each participant will have a name.

The concepts represent pairwise disjoint sets.
Case Study: Nokia Meetings

Representing the disjointness of the Concepts

<table>
<thead>
<tr>
<th>Ont_spec: nMeeting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topic</td>
</tr>
<tr>
<td>Location</td>
</tr>
<tr>
<td>Meeting</td>
</tr>
<tr>
<td>AgendaItem</td>
</tr>
<tr>
<td>Participant</td>
</tr>
<tr>
<td>Document</td>
</tr>
</tbody>
</table>

In DL, this would require 21 textual assertions, eg:

\[
\text{Topic} \sqcap \text{Meeting} \equiv \bot
\]
Case Study: Nokia Meetings

Roles

1. location: a role between Meeting (domain) and Location (codomain),
2. topic: Meeting (domain), Topic (codomain),
3. agenda: Meeting (domain), AgendaItem (codomain),
4. host: Meeting (domain), Participant (codomain),
5. participants: Meeting (domain), Participant (codomain),
6. documents: AgendaItem (domain), Document (codomain),
7. responsibilityOf: AgendaItem (domain), Participant (codomain),
8. name: Participant (domain), Name (codomain).
Case Study: Nokia Meetings

Representing roles

Ont_spec: nMeeting

[Diagram of a network of concepts related to meetings, including nodes for Location, Topic, AgendaItem, Document, Participant, Meeting, Name, and Participant, with edges indicating relationships such as location, topic, agenda, meetingHost, participants, and meetingHost.]
Case Study: Nokia Meetings

Constraints

if $a$ is an AgendaItem, on the agenda of Meeting $m$ and $a$ is the responsibilityOf Participant $p$ then it must be the case that the set of $m$’s participants includes $p$. 
if $a$ is an AgendaItem, on the agenda of Meeting $m$ and $a$ is the responsibilityOf Participant $p$ (above the line) then it must be the case that the set of $m$’s participants includes $p$ (below the line).
Case Study: Nokia Meetings

Queries

we may want to obtain the set of documents associated with the agenda items for some given meeting
Case Study: Nokia Meetings

we may want to obtain the set of documents associated with the agenda items for some given meeting

Ont_Query: Meeting getDocuments(m) → D
Case Study: Nokia Meetings

Instances

We can depict instances using constants.
We can reason at the diagrammatic level.
Case Study: Nokia Meetings

Mapping

Ont_spec: devAndPres
Case Study: Nokia Meetings

Mapping

<table>
<thead>
<tr>
<th>OntLink</th>
<th>OntSpec: nMeeting</th>
<th>OntSpec: devAndPres</th>
</tr>
</thead>
</table>

Meeting
  - nMeeting:Meeting
  - devAndPres:Meeting

Document
  - Papers
Case Study: Nokia Meetings

The set of documents associated with a meeting’s agenda items includes all of the papers associated with the meeting’s presentations.
Other Features of Ontology Diagrams

Ont_Constraint: myOntology Injective

\[ A \xrightarrow{f} a \quad \text{and} \quad A \xrightarrow{f} a \]

\[ x \equiv y \]
Other Features of Ontology Diagrams

Ont_Constraint: myOntology    MyRule

A

\( f \)

B

\( g \)

or

\( \)

\( x \)

\( x \)

\( g \)

\( f \)

\( x \)
Other Features of Ontology Diagrams

Ont_Query: myOntology \ f(x) \rightarrow S
Other Features of Ontology Diagrams

Ont_Spec: myOntology
Other Features of Ontology Diagrams

Ont_Spec: myOntology

A \quad g \quad B

C

x
Ontology Diagrams: Well Matchedness

Underlying Euler diagrams are well matched.
Ontology Diagrams: Well Matchedness

Underlying Euler diagrams are well matched.

Placing \( m \) inside Meeting is well matched: \( m \) is in the set of meetings.
Ontology Diagrams: Well Matchedness

Underlying Euler diagrams are well matched.

Placing m inside Meeting is well matched: m is in the set of meetings.

An arrow from m to its set of agenda items is well matched: arrows are directional representing a property of a binary relation.
Ontology Diagrams: Free Rides

We inherit their free rides.
Ontology Diagrams: Free Rides

We inherit their free rides.

Placing $m$ inside Meeting and asserting Meeting is disjoint from Participant gives, for free, $m$ is not a participant.

Also for free: $m$ is distinct from $p$.

Further free rides associated with arrows.
Conclusions and Future Work

1. Proposed a framework for ontology visualization
2. Explore more case studies to refine design
3. Formalize syntax and semantics
4. Consider possible tool support (including diagram layout)
5. Investigate relationships with DL and other formalisms
Thank you

www.cmis.brighton.ac.uk/research/vmg