Ontology Modeling

Pascal Hitzler
Data Semantics Laboratory (DaSe Lab)
Data Science and Security Cluster (DSSC)
Wright State University
http://www.pascal-hitzler.de
• EiCs: Pascal Hitzler
  Krzysztof Janowicz

• Funded 2010

• 2016 Impact factor of 1.786, top of all journals with “Web” in the title

• We very much welcome contributions at the “rim” of traditional Semantic Web research – e.g., work which is strongly inspired by a different field.

• Non-standard (open & transparent) review process.

• http://www.semantic-web-journal.net/
Forthcoming book

Motivation: EarthCube and GeoLink
EarthCube GeoLink Scenario

The NSF EarthCube Program:
Developing a Community-Driven Data and Knowledge Environment for the Geosciences

“concepts and approaches to create integrated data management infrastructures across the Geosciences.”

“EarthCube aims to create a well-connected and facile environment to share data and knowledge in an open, transparent, and inclusive manner, thus accelerating our ability to understand and predict the Earth system.”
GeoLink: An EarthCube “Building Block” project (2014-2017)

How to realize data search across many large-scale geoscience data repositories, such that

- The approach is extendable to new repositories.
- The scope can extend across all of the Geosciences.
- The search capabilities can be made more fine-grained in the future if desired.

Central idea: Use a modular, extendable ontology for the integration of metadata.
Modeling Patterns
ABoxes as graphs

SesameStreet has Actor JimHenson.
JimHenson hasName “Jim Henson”.

Diagram:
- SesameStreet
- has Actor
- JimHenson
- hasName
- “Jim Henson”
Problem!

SesameStreet has Actor JimHenson.
MuppetShow has Actor JimHenson.
JimHenson plays Kermit.
JimHenson plays Ernie.
JimHenson hasName “Jim Henson”.
Solution!

Diagram showing relationships between entities such as MuppetShow, Kermit, SesameStreet, Ernie, JH-MS-Role, JH-SSK-Role, JH-SSE-Role, and Jim Henson.
Schematically

```
SesameStreet  hasActor  JimHenson  hasName  "Jim Henson"
```

```
Movie  hasActor  Person  hasName  "Jim Henson"
```

```
Character  plays  ActorRole  performedBy  Person
```

```
Movie  hasActor  ActorRole  performedBy  Person
```

```
xsd:string  hasName
```

An **Ontology Design Pattern** (ODP) is a reusable successful solution to a recurrent ontology modeling problem.

[Gangemi 2005]
Axiomatization

\[ \top \sqsubseteq \forall \text{providesAgentRole}.\text{AgentRole} \]
\[ \text{AgentRole} \sqsubseteq \forall \text{performedBy}.\text{Agent} \]
\[ \exists \text{performedBy}.\text{Agent} \sqsubseteq \text{AgentRole} \]
\[ \text{AgentRole} \sqsubseteq \forall \text{startsAtTime}.\text{Timelinstant} \]
\[ \text{AgentRole} \sqsubseteq \forall \text{endsAtTime}.\text{Timelinstant} \]
\[ \text{AgentRole} \sqsubseteq \exists \text{providesAgentRole}^\neg . \top \]
\[ \text{AgentRole} \sqsubseteq =1 \text{performedBy}.\text{Agent} \]
\[ \text{AgentRole} \sqsubseteq =1 \text{startsAtTime}.\text{Timelinstant} \]
\[ \text{AgentRole} \sqsubseteq =1 \text{endsAtTime}.\text{Timelinstant} \]

\text{DisjointClasses}(\text{AgentRole}, \text{Agent}, \text{Timelinstant})
A Minimal Event Pattern

\[ \top \sqsubseteq \forall \text{hasSpatioTemporalExtent}.\text{SpatioTemporalExtent} \]
\[ \top \sqsubseteq \forall \text{providesAgentRole}.\text{AgentRole} \]
\[ \text{Event} \sqsubseteq \exists \text{hasSpatioTemporalExtent}.\text{SpatioTemporalExtent} \]
\[ \text{Event} \sqsubseteq \forall \text{subEventOf}.\text{Event} \]
\[ \exists \text{subEventOf}.\text{Event} \sqsubseteq \text{Event} \]
\[ \text{subEventOf} \circ \text{subEventOf} \sqsubseteq \text{subEventOf} \]

DisjointClasses(\text{Event, AgentRole, SpatioTemporalExtent})
Worked Example: Chess
Worked Example: Chess

• Establish a searchable repository for chess data.

• Starting point are PGN files.
  • Should be extendable with other information from
    – Chess websites
    – Wikipedia
    – Geographic data
    – News
    – Etc.

• Use an ontology for information integration.
GeoVoCamps modeling approach

- Collaborative modeling, group ideally has
  - More than one domain experts.
  - People familiar with the base data.
  - People understanding possible target use cases.
  - An ontology engineer familiar with the modeling approach.
  - Somebody who understands formal semantics of OWL.

- Domain experts are queried as to the main notions for the application domain.
  - E.g. for chess, these would include
    - Chess game; move; opening; tournament; players; commentary
GeoVoCamps modeling approach

• From available data and from application use cases, devise competency questions, i.e. questions which should be convertible into queries, which in turn should be answerable using the data.

1. Who played against Kasparov in the round 1994 Lineares tournament? Did (s)he play as a white or black player?
2. What is the first move taken by the black player in the Sicilian Defense opening?
3. Find all games in which Bobby Fischer, playing black, lost in the poisoned pawn variation of the Sicilian Defence opening.
4. Are there any recorded games using the Grünfeld Defence from before the 20th century?
5. What did Kasparov say about his opponent’s first two moves in his commentary about his game against Topalov in the 1999 Tournament in Wijk ann Zee?
6. Who was the first non-Russian world champion after Fischer?
7. Did Bobby Fischer ever play against a grandmaster in Germany?
8. List all world championship games won by forfeit.
Then prioritize which notions to model first. In the chess case, e.g.

- chess game
- move/half-move
- players
- opening
- tournaments
- commentary
GeoVoCamps modeling approach

- Understand the nature of the things you are modeling.

Chess game ... An Event
Half-move ... A Subevent of a chess game
Player ... The Role of an Agent
Opening ... this is probably complex
tournaments ... Events
commentary ... this is again more complex
Player as AgentRole

owl:Thing providesAgentRole AgentRole

AgentRole startedAtTime, endedAtTime TimeInstant

Agent

BlackPlayerRole rdfs:subClassOf AgentRole

ChessGame providesAgentRole AgentRole

WhitePlayerRole rdfs:subClassOf AgentRole

performedBy
ChessGame as Event

**Diagram:**
- **TemporalExtent** atTime **Event** providesAgentRole **AgentRole** performedBy **Agent**
  - subEventOf
  - atPlace **Place**

- **TemporalExtent** atTime **ChessGame** providesAgentRole **AgentRole** performedBy **Agent**
  - atPlace **Place**
Opening, game result, etc.

We call these “stubs”.

I.e. we’re aware that more fine-grained modeling will be needed for some use cases.

But currently there’s no reason to do it (not in use case, no data), so we only provide “hooks” for future development of the ontology.
Putting things together
Adding commentaries
Adequacy check

• Triplify sample data using the ontology. Does it work?

• Check if competency questions can be answered.

• Add axioms as appropriate (the graph is only for intuition, the OWL axioms are the actual ontology).

• (there are more post-hoc details to be taken care of, but let’s leave it at that)
Axioms in this case are mostly straightforward:

- Inherited from Event or AgentRole
- Scoped domain/range restrictions, possibly with some cardinalities
- Basic existentials
- Non-cyclicity of half-move sequence

What about adding, e.g., the following?

\[ \text{ChessGame} \sqsubseteq \geq 0 \text{subEventOf. ChessTournament} \]

If one of the roles of axiomatization is to improve human understanding of the ontology, then such axioms are helpful!
Shortcuts and Views
Shortcuts

ChessGame(x) ∧ pAR(x, y) ∧ WhitePlayerRole(y) ∧ performedBy(y, z) ∧ Agent(z) ∧ hasName(z, s) → hasWhitePlayer(x, s)

ChessGame(x) ∧ pAR(x, y) ∧ BlackPlayerRole(y) ∧ performedBy(y, z) ∧ Agent(z) ∧ hasName(z, s) → hasBlackPlayer(x, s)
Translating the rules

However note that the introduction of additional role chains may cause violations of regularity restrictions.
Simplified View

- ChessGame
  - atChessTournament
  - atPlaceNamed
  - atTime
  - subEventOf
  - hasOpeningECO
  - hasReport
  - hasFirstHalfMove, hasLastHalfMove, hasHalfMove
  - playedBy
  - hasAuthor
  - hasWhitePlayer, hasBlackPlayer
  - hasResultSAN
  - hasPGNFile

- HalfMove
  - nextHalfMove

- xsd:string
  - hasSANRecord

- xsd:dateTime
Mapping from Views

We used rules (axioms) to express the mapping from the ontology to the view.

The reverse direction is much more tricky.

ClassA(x) \land ClassB(y) \land C_1(x_1) \land \cdots \land C_n(x_n) \land R_1(y_1, y_2) \land \cdots \land R_k(y_k, y_{k+1})

\rightarrow \text{shortcut}(x, y).

\text{shortcut}(x, y) \rightarrow \text{ClassA}(x) \land \text{ClassB}(y) \land \exists x_1 \ldots \exists x_n \exists y_1 \ldots \exists y_n (C_1(x_1) \land \ldots \land C_n(x_n) \land R_1(y_1, y_2) \land \cdots \land R_k(y_k, y_{k+1}))
Mapping from views

Existential rules may be suitable in principle.

\[
\text{shortcut}(x, y) \rightarrow \text{ClassA}(x) \land \text{ClassB}(y) \land \exists x_1 \ldots \exists x_n \exists y_1 \ldots \exists y_n (C_1(x_1) \land \ldots \\
\ldots \land C_n(x_n) \land R_1(y_1, y_2) \land \ldots \land R_k(y_k, y_{k+1}))
\]

However automated reasoning with the potentially rather complex rule heads requires investigations, in particular if it is to be integrated with ontology reasoning.

A specific case are right-hand-side role chains:

\[
R \sqsubseteq R_1 \circ \cdots \circ R_n,
\]
The GeoLink Modular Ontology (GMO)
Back to GeoLink
High-level overview of the GeoLink Modular Ontology (GMO). Each box stands for a module, which has been modeled in its own right.
Example Module: Cruise

Cruise reused e.g. the generic patterns
AgentRole
Trajectory
and conceptually cruises are understood to be events.
An (preliminary) interactive demonstration of the integrated GeoLink data is available at

http://demo.geolink.org

At http://www.geolink.org/ there are links to the complete schema, a SPARQL Endpoint, publications, etc.
Several W3C recommendations are relevant in our context:

- The Web Ontology Language OWL for expressing ontologies and sharing them on the Web.

- The Resource Description Framework RDF for expressing e.g. ABox graphs and aligning them to an ontology.

- The SPARQL RDF query language.

(See the references for pointers.)
Thanks!
References


References


Adila Krisnadhi, Ontology Pattern-Based Data Integration. Dissertation, Department of Computer Science and Engineering, Wright State University, 2015.
References


References


Steve Harris, Andy Seaborne, SPARQL 1.1 Query Language. W3C Recommendation 21 March 2013.