Some patterns and their use in the chess ontology

Pascal Hitzler
Data Semantics Laboratory (DaSe Lab)
Data Science and Security Cluster (DSSC)
Wright State University
http://www.pascal-hitzler.de
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.
GeoVoCamps modeling approach

- 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.

<table>
<thead>
<tr>
<th>Chess game</th>
<th>...</th>
<th>An Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Half-move</td>
<td>...</td>
<td>A Subevent of a chess game</td>
</tr>
<tr>
<td>Player</td>
<td>...</td>
<td>The Role of an Agent</td>
</tr>
<tr>
<td>Opening</td>
<td>...</td>
<td>this is probably complex</td>
</tr>
<tr>
<td>tournaments</td>
<td>...</td>
<td>Events</td>
</tr>
<tr>
<td>commentary</td>
<td>...</td>
<td>this is again more complex</td>
</tr>
</tbody>
</table>
Player as AgentRole

owl:Thing \rightarrow AgentRole
- providesAgentRole

AgentRole \rightarrow TimeInstant
- performedBy
- startsAtTime, endsAtTime

BlackPlayerRole \rightarrow AgentRole
- rdfs:subClassOf

WhitePlayerRole \rightarrow AgentRole
- rdfs:subClassOf

ChessGame \rightarrow AgentRole
- providesAgentRole

AgentRole \rightarrow Agent
- performedBy
ChessGame as Event

Diagram showing the relationship between TemporalExtent, Event, AgentRole, Agent, Place, and ChessGame, with arrows indicating the relationships: atTime, providesAgentRole, performedBy, and subEventOf.
Half-moves
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?

ChessGame $\sqsubseteq \exists 0 \text{subEventOf}.\text{ChessTournament}$

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

\[
\text{ChessGame}(x) \land \text{pAR}(x, y) \land \text{WhitePlayerRole}(y) \land \text{performedBy}(y, z) \\
\land \text{Agent}(z) \land \text{hasName}(z, s) \rightarrow \text{hasWhitePlayer}(x, s) \\
\text{ChessGame}(x) \land \text{pAR}(x, y) \land \text{BlackPlayerRole}(y) \land \text{performedBy}(y, z) \\
\land \text{Agent}(z) \land \text{hasName}(z, s) \rightarrow \text{hasBlackPlayer}(x, s)
\]
Translating the rules

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)

ChessGame ⊑ ∃R₁.Self
WhitePlayerRole ⊑ ∃R₂.Self
Agent ⊑ ∃R₃.Self

R₁ ◦ pAR ◦ R₂ ◦ performedBy ◦ R₃ ◦ hasName ⊑ hasWhitePlayer

However note that the introduction of additional role chains may cause violations of regularity restrictions.
Modeling OWL with Rules (ROWLTab)

• Protégé Plug-In

• Enter rules using interface very similar to SWRLTab.
• But rules are converted into OWL axioms (whenever possible) instead of DL-safe rules.

E.g., \( \text{Pig}(x) \rightarrow \text{Mammal}(x) \) becomes \( \text{Pig} \sqsubseteq \text{Mammal} \) and thus carries the correct semantics.

http://dase.cs.wright.edu/content/modeling-owl-rules

We evaluated that ROWL leads to quicker modeling with fewer errors.

http://dase.cs.wright.edu/content/rowl
And see full paper here at ESWC2017
Simplified View
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 \cdots \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.

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

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:

\[ R \sqsubseteq R_1 \circ \cdots \circ R_n, \]
Thanks!
References


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


References
