An Ontology Design Pattern for Chess Games

Adila Krisnadhi[1,2], Víctor Rodríguez-Doncel[3], Pascal Hitzler[1], Michelle Cheatham[1], Nazifa Karima[1], Reihaneh Amini[1], and Ashley Coleman[1]

[1] Data Semantics Lab, Wright State University
[2] Faculty of Computer Science, Universitas Indonesia
[3] Ontology Engineering Group, Universidad Politécnica de Madrid

Abstract. We present an ontology pattern describing records of chess games. Besides being an interesting modeling problem by itself, the fact that chess is one of the most popular game in the world with hundreds of millions of active players, including several millions online players led to a huge amount of chess game data available from various online chess databases. Furthermore, these data are becoming available as linked data with links to prominent datasets, such as DBPedia or GeoNames. However, there is still a lack of a well-designed schema that can ease linked data publishing of chess games. Our pattern is hoped to fill that gap.

1 Introduction

Chess is one of the most popular game played by people worldwide. A 2012 study found that "605 million adults play chess regularly - a number comparable to regular users of Facebook." [1]. Among them, millions of players actively play chess online, e.g., Chess.com boasts more than 12 million members with tens of thousands players play concurrently online [2]. More interestingly, the large number of chess games played online led to huge repositories of chess games that people can access in the form of Portable Game Notation (PGN) files. Chess game data does not just contain details about chess moves, but also about player identities, chess tournament names, and spatiotemporal information relevant to the games. Making these data available also as *linked data* is an interesting proposal as we can generate even richer information by combining chess game data with data from prominent linked datasets, such as DBPedia, GeoNames, or Library of Congress. We have described an effort to realize this proposal elsewhere [5]. What is lacking, however, is a well-designed schema for these data that is still sufficiently flexible and extendible in the long run. Thus, to serve this purpose, we present an ontology design pattern for chess games, engineered by following a modular approach.

The design of the pattern is mainly guided by the PGN file format[1] because this format is the de-facto, computer-processable, standard format for recording chess games and supported by many chess programs. Each game description in a PGN file contains a tag-pair section and a movetext section. The latter describes

[1] `http://www.thechessdrum.net/PGN_Reference.txt`

the actual moves in the game using the Standard Algebraic Notation (SAN), possibly with additional annotation, while the former provides information in the form of name-value pairs expressed as a tag enclosed with square brackets. The tag-pair section contains seven mandatory information, namely the name of tournament or match event, the location of the event, the starting date of the game, the playing round for the game in a tournament, the white player's name, the black player's name, and the result of the game. Additionally, it may also contain other information such as players' Elo rating, opening variation, alternative starting position, or description of the game conclusion. As a file format, PGN allows one to record almost any relevant information of a chess game. However, the format does not permit us, at least in a straightforward way, to incorporate web-style of data linking. Moreover, querying specific piece of data from a PGN file requires custom-built parser. Thus, to help the design process, we conceived a few examples of competency questions such as: (i) "Who played against Kasparov in the round 1994 Linares tournament? Did (s)he play as a white or black player?"; (ii) "What is the first move taken by black player in the Sicilian Defence opening?"; (iii) "List all moves in a Fool's Mate game where black wins after 2 moves by both players?"; and (iv) "What did Kasparov say about his opponent first two moves in his commentary about his game against Topalov in 1999 Tournament in Wijk aan Zee?"

2 The Chess Pattern

The Chess pattern[2], depicted in Fig. 1, consists of a core part that models chess games as a series of half-moves performed by chess players, and a few additional parts, modeling the game annotation, opening, result, and the structure of chess tournament as abstracted from FIDE (Federation Internationale des Echecs) regulations[3]. The concise DL notation [4] is used to express the axioms below.

Chess Game. The ChessGame class represents chess games, modeled as a kind of event (Axiom 2), which itself must happen at some place and time (Axiom 1). Further, Axiom 2 also asserts that a chess game involves actors performing some role, e.g., white and black players. The TemporalExtent and Place classes are hooks for attaching spatial and temporal information to a chess game, whose model is out of scope of this paper. Actors in a chess game are agents who perform some role provided by the game. Instead of employing binary relationships using properties such as "hasWhitePlayer", we employ reification via the AgentRole class. From this complication, we gain flexibility and the ease of adding contextual information when necessary, e.g., title or Elo rating of the player when the game was played. Every chess game provides at least two distinct role instances, namely for the white and the black players, as ensured by Axiom 2 – where pAR is a shorthand for providesAgentRole – and the disjointness between WhitePlayerRole and BlackPlayerRole explained at the end of this section. Each agent role is performed by exactly an agent as asserted by Axiom 3. Agents can

[2] `http://ontologydesignpatterns.org/wiki/Submissions:ChessGame`
[3] `http://www.fide.com/fide/handbook.html`

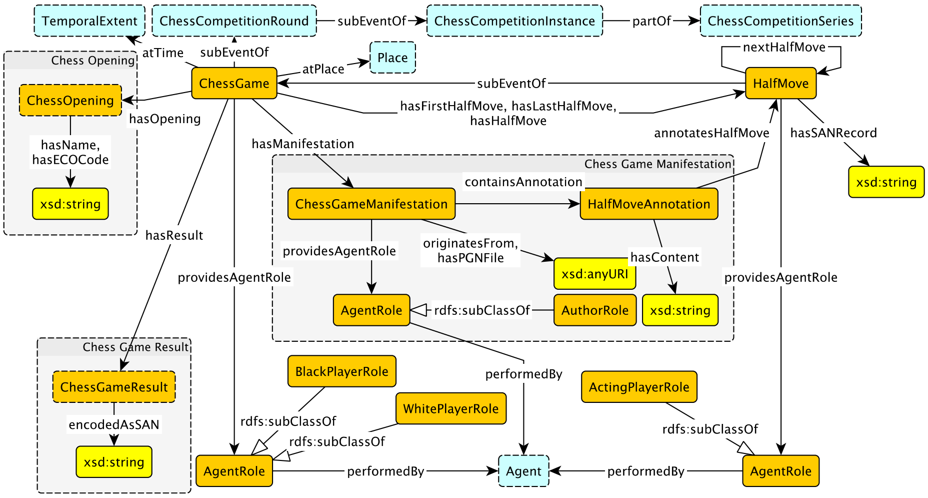

Fig. 1. The core of Chess pattern modeling chess games. Though Event is omitted, half moves, chess games, and chess competition rounds/instances/series are events. Orange boxes are atomic classes. Blue boxes are classes with details hidden in a separate pattern/ontology. Grouping boxes are sub-patterns that can be modeled separately.

be a person, a team, a computer software, etc. Note that the axioms still allow one agent to perform two roles simultaneously. Also, we allow a chess game played by multiple white players and/or multiple black players. On the other hand, Axiom (4) asserts that instances of WhitePlayerRole and BlackPlayerRole are roles specific for a particular ChessGame.

$$\mathsf{Event} \sqsubseteq \exists \mathsf{atPlace.Place} \sqcap \exists \mathsf{atTime.TemporalExtent} \tag{1}$$

$$\mathsf{ChessGame} \sqsubseteq \mathsf{Event} \sqcap \exists \mathsf{pAR.BlackPlayerRole} \sqcap \exists \mathsf{pAR.WhitePlayerRole} \tag{2}$$

$$\mathsf{AgentRole} \sqsubseteq (=1\ \mathsf{performedByAgent}) \tag{3}$$

$$\mathsf{WhitePlayerRole} \sqcup \mathsf{BlackPlayerRole} \sqsubseteq \mathsf{AgentRole} \sqcap (=1\ \mathsf{pAR}^{-}.\mathsf{ChessGame}) \tag{4}$$

We model the standard chess rule where each player takes turn performing a *half move* by moving one of his pieces. A finite sequence of half-moves constitutes a chess game, which has exactly one first half-move and one last half-move (Axiom 5). The axioms, however, do not enforce that a chess game has to start from the initial chessboard configuration, nor they assert that the white player is the one who moves first. This allows the pattern to model, e.g., chess problems and their solutions. Half-moves are represented by the HalfMove class, itself a kind of event that is a subevent of exactly one chess game, and provides a role for the corresponding acting player (Axiom 6 and 7). Axiom 8 states that a half-move can only be followed by exactly one other half-move and cannot follow itself. Also, a half-move may have an associated record in SAN given as a string value.

$$\mathsf{ChessGame} \sqsubseteq (=1\ \mathsf{hasFirstHalfMove.HalfMove}) \sqcap (=1\ \mathsf{hasLastHalfMove.HalfMove}) \quad (5)$$

$$\mathsf{HalfMove} \sqsubseteq \mathsf{Event} \sqcap \exists \mathsf{pAR.ActingPlayerRole} \sqcap (=1\ \mathsf{hasHalfMove}^{-}.\mathsf{ChessGame}) \quad (6)$$

$$\mathsf{hasHalfMove} \sqsubseteq \mathsf{subEventOf}^{-}, \mathsf{hasFirstHalfMove} \sqcup \mathsf{hasLastHalfMove} \sqsubseteq \mathsf{hasHalfMove} \quad (7)$$

$$\mathsf{HalfMove} \sqsubseteq (\leqslant 1\ \mathsf{nextHalfMove.HalfMove}) \sqcap \neg \exists \mathsf{nextHalfMove.Self} \quad (8)$$

The ChessGameResult class captures information about the result of a chess game, which may be encoded in SAN. A chess game may also be associated with a standard chess opening from the Encyclopedia of Chess Opening (ECO).[4] In addition, a chess game may be a subevent of a chess competition round and may have a chess game manifestation as explained below.

Chess Competition. Important notions here are *round*, *tournament instance*, and *tournament series*, respectively represented by the Chess Competition Round, Chess Competition Instance, and Chess Competition Series sub-patterns. They are all modeled as events, so we simply assert:

$$\mathsf{ChessCompetitionRound} \sqcup \mathsf{ChessCompetitionInstance} \sqcup \mathsf{ChessCompetitionSeries} \sqsubseteq \mathsf{Event}$$

An example of chess competition series is the Linares International Chess Tournament, while its 1994 edition is an example of a chess competition instance. Here, a chess competition instance is modeled as a part of a chess competition series, instead of the subevent relationship since the latter may a partonomic relationship regarding their spatial dimension (see, e.g., Eq. 2.2 in [3]).

Chess Game Manifestation. From Fig. 1, a chess game may have a chess game manifestation, represented by the ChessGameManifestation class that may provide roles, e.g., author roles for some agents – the existence of an agent for a given agent role was asserted in Axiom 3. Note that chess game manifestations are not events, but rather creative works by humans. A chess game manifestation may originate from some external source or PGN file indicated by a URI. A chess game manifestation may have annotations for half-moves containing comments given as a string. We leave a complete schema for all possible structures of the content of a PGN file for future work. So, we assert:

$$\mathsf{AuthorRole} \sqsubseteq \mathsf{AgentRole} \sqcap \exists \mathsf{pAR}^{-}.\mathsf{ChessGameManifestation} \quad (9)$$

$$\mathsf{HalfMoveAnnotation} \sqsubseteq \exists \mathsf{annotatesHalfMove.HalfMove} \sqcap \exists \mathsf{hasContent.xsd{:}string} \quad (10)$$

We *can* also assert that half-moves annotated in a chess game manifestation have to be the half-moves of the corresponding chess games as in the following axiom, which may cause the pattern to go beyond OWL 2 if combined with Axiom 6.

$$\mathsf{hasManifestation} \circ \mathsf{containsAnnotation} \circ \mathsf{annotatesHalfMove} \sqsubseteq \mathsf{hasHalfMove}$$

Class Disjointness, Domain and Range Restrictions. We assert class disjointness for every pair of classes in this pattern, except when the pair contains two classes for which a subclass axiom is explicitly asserted. We also assert

[4] `http://www.chessinformant.org/eco-encyclopedia-of-chess-openings`

guarded domain and range restrictions for every property in the pattern, e.g., for hasHalfMove, this would be in the form of:

$$\exists \mathsf{hasHalfMove}.\mathsf{HalfMove} \sqsubseteq \mathsf{ChessGame}, \quad \mathsf{ChessGame} \sqsubseteq \forall \mathsf{hasHalfMove}.\mathsf{HalfMove} \quad (11)$$

Revisiting the examples of competency questions at the end of Section 1, we can see that Question (i) can be answered because the pattern models involvement of agents in a chess game, as well as captures chess tournament through chess competition round/instance/series. Question (ii) can be answered as the pattern allows us to model the first few moves of any game where the Sicilian Defence opening is used. Answers to Question (iii) can be obtained if the so-called Fool's Mate game can be described by the pattern, and the pattern can indeed describe a complete list of moves in any chess game. Finally, Question (iv) refers to commentary about a particular game, which is facilitated through instantiating the corresponding chess game manifestation.

3 Conclusion

We have presented an ontology pattern for chess games. This pattern, on the one hand, coves key notions understood in chess playing community. Some parts of the pattern were left underspecified as modeling them in full would require separate patterns and more in-depth analysis. As the next step, we will investigate how this pattern would fare in the context of linked data publishing of chess games data that are widely available online.

Acknowledgements. This work was partially supported by the National Science Foundation under award 1440202 *EarthCube Building Blocks: Collaborative Proposal: GeoLink – Leveraging Semantics and Linked Data for Data Sharing and Discovery in the Geosciences*, by the European Commission under the International Reseach Staff Exchange Scheme (IRSES) of the EU Marie Curie Actions, project *SemData – Semantic Data Management*, and by the Spanish Ministry of Economy and Competitiveness (project TIN2013-46238-C4-2-R).

References

1. Chess Redux: how many people play chess? `http://en.chessbase.com/post/che-redux-how-many-people-play-che-` (August 2012), last accessed: July 7, 2015
2. Chess.com. `http://www.chess.com` (Last accessed: July 7, 2015)
3. Gupta, A., Jain, R.: Managing Event Information: Modeling, Retrieval, and Applications: Modeling, Retrieval, and Applications. Synthesis Lectures on Data Management, Morgan & Claypool Publishers (2011)
4. Krötzsch, M., Simančík, F., Horrocks, I.: A description logic primer. CoRR abs/1201.4089 (2012)
5. Rodríguez-Doncel, V., Krisnadhi, A., Hitzler, P., Cheatham, M., Karima, N., Amini, R.: Pattern-based linked data publication: The linked chess dataset case. Tech. rep., Data Semantics Lab, Wright State University (2015), available from `http://dase.cs.wright.edu/publications/pattern-based-linked-data-publication-linked-chess-dataset-case`