

Semantic Chess

Handbuch

Inhaltsverzeichnis

1. Einleitung.....	3
2. Installation.....	3
3. Strukturierung.....	3
3.1 Home.....	4
3.2 Projektverlauf.....	4
3.3 Downloads.....	4
3.4 Wiki.....	4
3.5 Team.....	4
3.6 Demo.....	4
4. Funktionen.....	5
4.1 Konvertierung.....	5
4.2 Suche.....	6
4.3 Spielen	8
5. Wesentliches für Administratoren.....	9
5.1 Konvertierung großer Datenbestände.....	9
5.2 TripleStore-Zugriff, Verwaltung der Datenbank.....	10
5.3 Datenlinking.....	10
6. Weitere Informationen.....	10

1. Einleitung

Seit Jahrzehnten wird immer wieder mit neuen technologischen Voraussetzungen versucht innovative, intelligente Systeme für Schachspiele zu entwickeln. Ein neuer Ansatz ist die Nutzung der Technologie „Semantic Web“, welche auch in unserer Software Anwendung findet.

Das folgende Handbuch dient zur erleichterten Bedienung des Softwareproduktes „Semantic Chess“. Hierbei wird genauer auf die einzelnen Funktionen des Produktes eingegangen und diese werden Schritt für Schritt erläutert.

2. Installation

Falls durch ein Folgeprojekt (o.ä.) eine Installation erforderlich ist, ist folgende Software zu fokussieren: **Apache Tomcat** mit **Vaadin** und **Virtuoso**.

Die Installationsanweisungen für diese Software sind online zu finden, falls aber spezielle Probleme auftreten, gehen die Projektmitglieder gern auf Fragen ein.

Es sei aber gesagt, dass die Konfiguration nicht für jedermann sonderlich trivial erscheinen.

3. Strukturierung

Die Gesamtheit aller Projektdaten sind auf der Homepage :

<http://pcai042.informatik.uni-leipzig.de/~swp13-sc/>

zu finden. Sie unterteilt sich in 6 verschiedene Tabs:

- Home
- Projektverlauf
- Downloads
- Wiki
- Team
- Demo

3.1 Home:

Nach Betätigung des Home – Tabs, erscheint eine kurze Projektvorstellung und es werden die verwendeten Technologien aufgelistet. Bei Fragen oder Anregungen können die Nutzer über die E-Mail Adresse das Projektteam erreichen.

3.2 Projektverlauf:

Unter dem Tab Projektverlauf kann der Nutzer sich einen Überblick über die Entwicklungshistorie verschaffen.

3.3 Downloads:

Hier können die gesamten Dokumente, die das Projektteam während des Praktikums erstellt hat, heruntergeladen werden.

3.4 Wiki:

Das bisher unausgereifte Wiki bietet eine Wissensbasis bzw. Wissensansammlung von durch das Projekt angeeigneten Erkenntnissen.

3.5 Team:

Das Einsehen der Aufgabenverteilungen im Zusammenhang der verschiedenen Projektmitglieder ist auf diesem Tab möglich. Wobei der Stellvertreter nicht außer Acht gelassen wurde.

3.6 Demo:

Hier ist das eigentliche Produkt zu finden. Die Benutzeroberfläche ist in zwei zu durch den mittleren Balken horizontal regulierende Bereiche unterteilt. Der linke Bereich umfasst das Such- und Konvertierungsinterface. Der rechte Bereich beinhaltet die grafische Benutzeroberfläche der Caballo – Engine.

4. Funktionen

Die Funktionen sind in folgende drei Unterpunkte unterteilt:

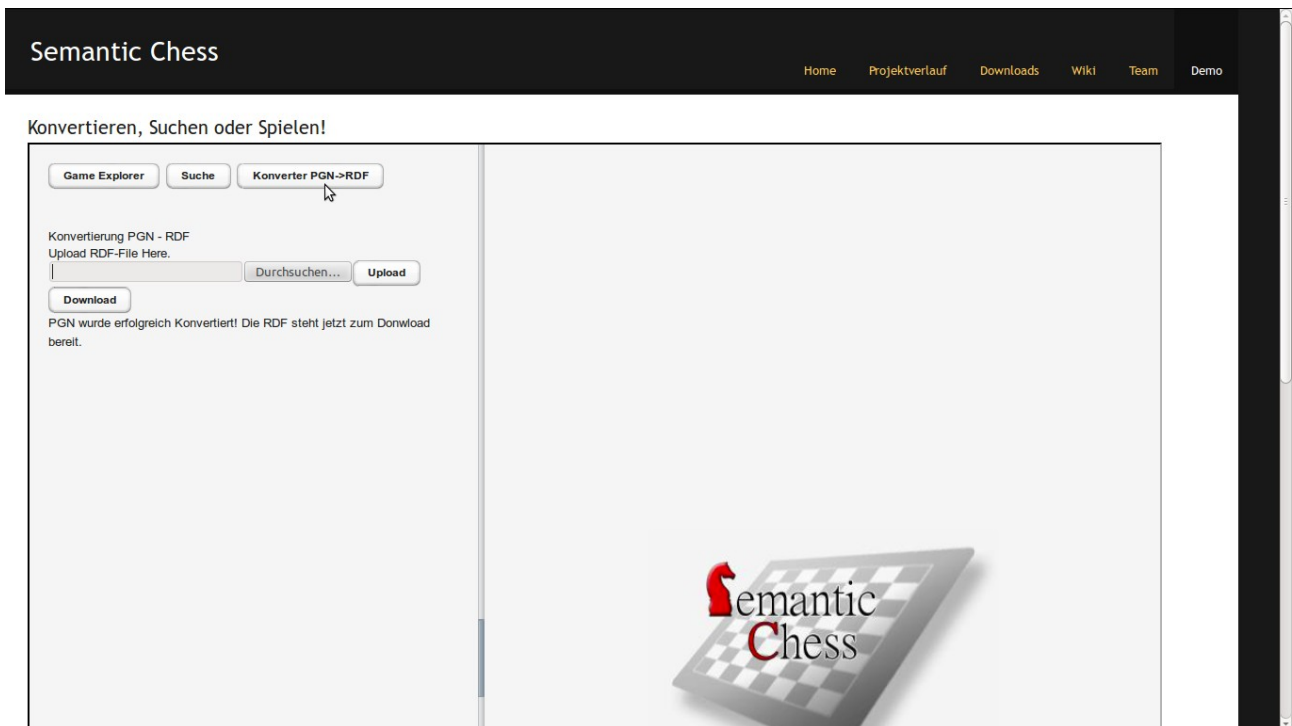
- Konvertierung
- Suche
- Spielen

Die Schritte interagieren und bedingen teilweise einander, was zu Überschneidung der einzelnen Funktionalitäten führen kann. Es empfiehlt sich daher, alle Funktionen zu verinnerlichen, um die volle Mächtigkeit der Software anzuwenden.

4.1 Konvertierung

Eines der Features, die das Produkt liefert, ist die Konvertierung von PGN (Portable Game Notation) zu RDF (Resource Description Framework). Die Nutzung dieser Funktion ist sehr einfach gehalten:

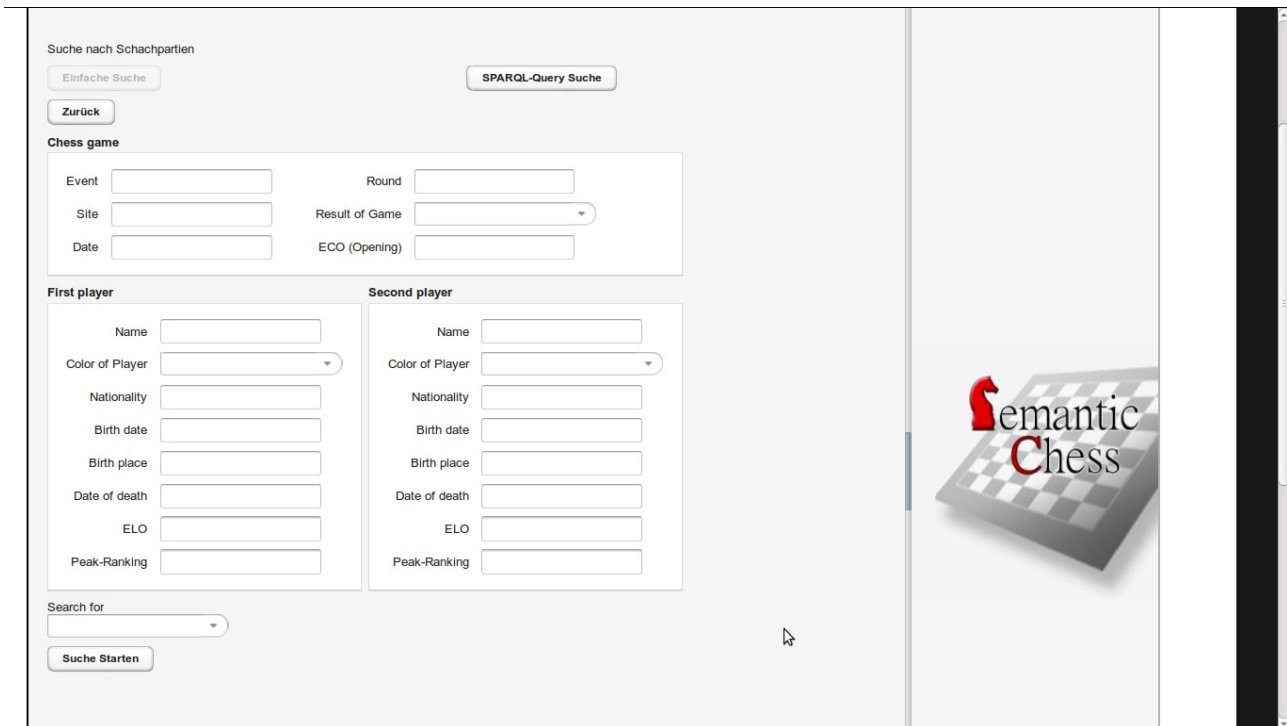
Man betätigt den dritten Button „Konverter PGN ->RDF“. Somit öffnet sich ein in die Oberfläche integriertes Interface, wo man lokal eine PGN- Datei auswählen kann, welche dann nach dem Drücken des „Upload“-Button auf den Server gespeichert werden, wo die Daten konvertiert werden. Durch Drücken des „Download“-Buttons kann die zu RDF konvertierte Datei herunter geladen werden.



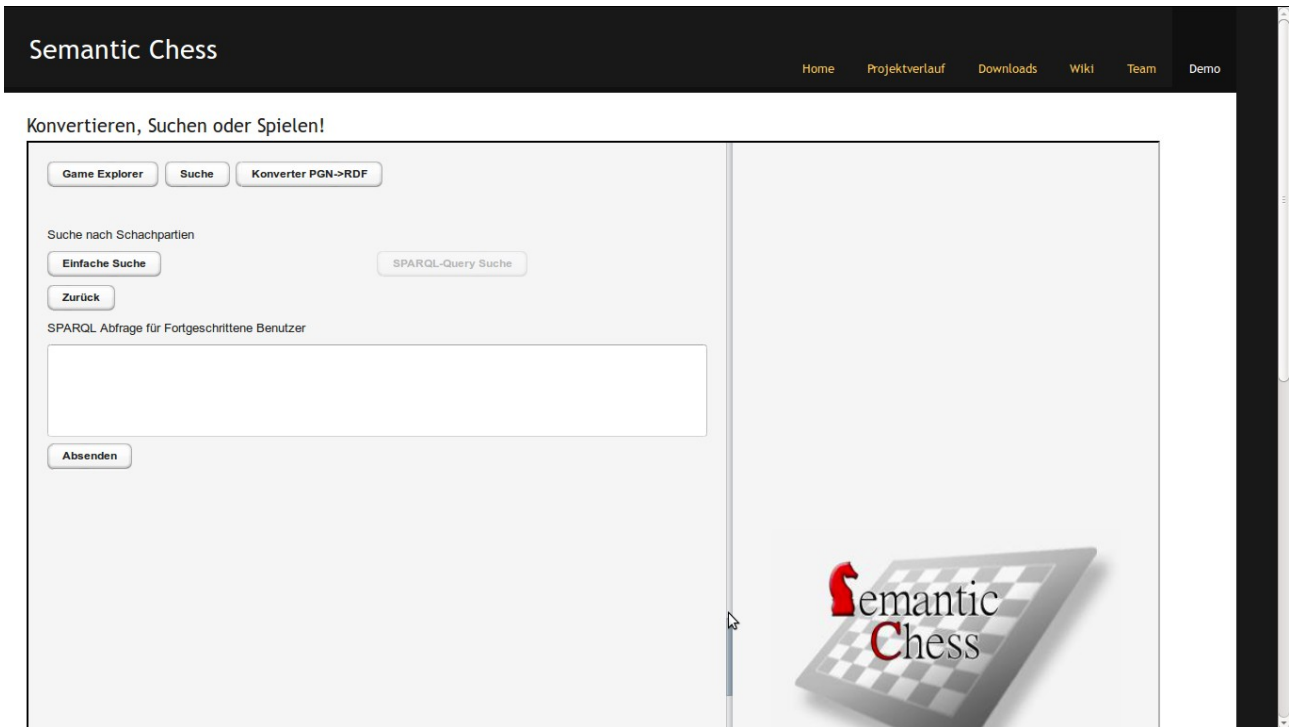
4.2 Suche

Die durch die Uploads der Administratoren erstellte RDF-Datenbank kann über „Suche“ abgefragt werden.

Abfragen kann man auf 2 verschiedene Möglichkeiten : „Einfach Suche“ und „SPARQL-Query Suche“. Die einfach Suche ermöglicht eine Auswahl mehrerer Optionen (z.B. Schachspielausgang, Spielernamen, ...), welche nach Betätigung des Button eingeblendet werden.



Auf der anderen Seite kann man aber auch eine explizite SPARQL – Abfrage erstellen, bei der man durch spezielle SPARQL-Syntax benutzerspezifische Suchkriterien angeben kann. Dies ist auch direkt über den SPARQL-Endpoint möglich zu finden unter: <http://pcai042.informatik.uni-leipzig.de:1358/sparql>



Durch das Abfragen der Daten, wird eine begrenzte Anzahl an Spielen (oder allgemein Daten) ausgegeben, welche unterhalb des Suchfensters angezeigt werden.

4.3 Spielen

Nach der Suche kann auf eines der gefundenen Ergebnisse geklickt werden. Somit wird die URI kopiert und bietet nach Betätigung der Button „Replay Game“ für Caballo eine Referenz zu dem Spiel. Dadurch kann man das Spiel nachspielen und bei jedem Spielzug überlegen, wie man selbst vorgehen würde.

The screenshot displays a web-based chess interface. On the left, there is a search form with fields for 'Site' and 'Date', and a 'First player' section with fields for 'Name', 'Color of Player' (set to 'white'), 'Nationality', 'Birth date', 'Birth place', 'Date of death', 'ELO', and 'Peak-Ranking'. Below the search form is a list of search results, each with a URI. The main area shows a chessboard with pieces in their starting positions. The board is labeled with columns 'a' through 'h' and rows '1' through '8'. The square b2 is highlighted in yellow. To the right of the board is a 'Game info' panel with 'Current player: White', 'back' and 'forward' buttons, and a 'Replay Game' button. Below this is a 'History' list of moves: 12. d5, 13. Re1, 14. Ne4, 15. c3, 16. f5, 17. Nh4, 18. Nf6, 19. Ob3, 20. Nbd7, 21. e4, 22. fxe4, 23. Nxe4, 24. Nxe4, 25. Bxe4, 26. Nf6, 27. Bg2, 28. Ng4, 29. Bf4, 30. e6, 31. Nf3, 32. h6, 33. h3, 34. Nf5, 35. Be5, 36. Kd7. Below the board is a text box for 'Spiel-URI hier eingeben:' with a 'Replay Game' button. At the bottom, there is a table of game details:

J. Briancon vs. Didier	Ergebnis: 0-1	ECO: D02
Event: Chanac, Runde: 1	Datum: 1989.?? ??	Ort: Chanac
J. Briancon vs. Didier	Ergebnis: 0-1	ECO: D02
Event: Chanac, Runde: 1	Datum: 1989.?? ??	Ort: Chanac

Neben der Möglichkeit die Spiele nachzuempfinden, werden weiterhin alle Eckdaten zum Spiel angegeben, die man sonst auch hätte in der PGN nachlesen können.

Falls ein anderes Spiel im Präferenzfokus liegt, kann man einfach eine andere URI wählen und wieder die „Replay Game“ - Buttons drücken.

5. Wesentliches für Administratoren

Für Administratoren sind einige zusätzliche Informationen relevant, die hier erläutert werden.

5.1 Konvertierung großer Datenbestände

Die Konvertierung der PGN-Dateien mit den Schachspielen in RDF-Tripel lässt sich neben der Upload-Funktion der Web-Applikation auch lokal mit einem Konsolenprogramm durchführen. Dabei stehen eine Reihe von Optionen zur Verfügung, welche die Konvertierung beschleunigen können und auch eine Kompression für große Ein-/Ausgabedateien bieten.

Der Konverter lässt sich wie folgt starten: Im Verzeichnis mit der **PGNConverter.jar**, dem Konsolenprogramm zur Konvertierung PGN in RDF, die Kommandozeile öffnen. Dort lässt sich mit „`java -jar PGNConverter.jar`“ der Konverter starten, der automatisch die Hilfe anzeigt (auch mit „`--help`“ möglich).

Mit „`--compresslevel=<0-9>`“ lässt sich die Kompressionsrate steuern, die der Konverter bei der Erstellung der ZIP-Datei mit dem Ausgabeergebnis nutzt (0 stellt keine Kompression dar und 9 entsprechend die höchste). Ausgeschaltet kann die automatische Kompression mit „`--dontcompress`“.

Werden neben dem Standard-Ausgabeformat „`TURTLE`“ andere gewünscht, so müssen diese mit „`--format=<format>`“, wie auch in der Programmhilfe angegeben, spezifiziert werden. Es stehen noch „`N-TRIPLE`“ (Grundform von Turtle, wird nicht empfohlen) und „`RDF/XML`“ sowie „`RDF/XML-ABBREV`“ zur Verfügung. „`RDF/XML-ABBREV`“ stellt eine Verbesserung zu „`RDF/XML`“-Format dar, da hier Prefixe verwendet werden können, die die Ausgabe etwas komprimieren. Der letzte Parameter „`--split=<number>`“ wird wohl einer der mächtigsten sein. Er dient dazu, die Ausgabe in die in „`number`“ angegebene Anzahl von Schachspielen zu splitten, die bei eingeschalteter Kompression alle in einem ZIP-Archiv gespeichert werden. Je nach gewählter Größe schwanken dadurch die Zeit für die Konvertierung und auch die Ausgabedateigröße bei Kompression. Für Computer, die Java 500-1000MB Speicher bieten, wird eine Splitrategie von 500 empfohlen, darüber hinaus ist 1000 auch gut. Eine höhere Rate wird nicht empfohlen, da dabei das Schreiben der Ausgabedateiteile zu lange dauert. Mit einer Rate von 1 lassen sich somit auch Schachspiele einzeln konvertieren und abspeichern. Eine Rate von 0 bedeutet überhaupt keine Splittung, was nur für PGNs mit weniger als 10000 Spielen empfohlen wird, darüber hinaus wird es ineffizient!

Die Angabe der zu konvertierenden Dateien erfolgt einfach durch Angabe der Dateinamen oder einer Maske wie z. B. „`*.pgn`“ nach oder zwischen den anderen Optionen. Es sollte dabei darauf geachtet werden, dass die Eingabedateien nicht mit „`--`“ anfangen und auch, dass die Ausgabedateien den gleichen Namen wie die Eingabedateien haben, jedoch mit der Endung „`.ttl`“, „`.rdf`“, „`.nt`“ (formatabhängig) statt „`.pgn`“ oder noch „`.zip`“, falls mit Kompression. Bestehende Dateien (aus vorherigen Konvertierungen der gleichen Eingabedatei) werden somit ohne Nachfrage überschrieben!

5.2. TripleStore-Zugriff, Verwaltung der Datenbank

Die Verwaltung der Datenbank erfolgt durch Virtuoso (in unserem Fall erreichbar unter: <http://pcai042.informatik.uni-leipzig.de:1358/conductor/>). Die Anmelde-Daten sind standardmäßig auf Nutzernamen: dba, Passwort: dba gesetzt. Dort können alle Möglichkeiten, die die verwendete Virtuoso-Version bietet, genutzt werden. Unter dem Tab „RDF“ finden sich RDF-spezifische Operationen. Unter SPARQL können wie im SPARQL-Endpoint Anfragen gestellt werden, unter „RDF Store Upload“ können Daten der RDF-Datenbank hinzugefügt werden.

5.3 Datenlinking

Das Projekt bietet die Möglichkeit den aktuellen Datenbestand mit dem der DBPedia abzugleichen um ein Linking durchzuführen, um nicht in PGN enthaltene Informationen über die Spieler zu erhalten (z.B. Geburtsdaten, Elo, Titel, etc.). Das Linking wird gestartet, indem in ein entsprechendes Verzeichnis gewechselt wird, wo sich die Linker.jar, die sc1.xml und die limes.dtd befinden. Mit einem Konsolenaufruf von **java – jar Linker.jar** erfolgt das eigentliche Linking. Mit der **sc1.xml** lassen sich die Einstellungen für das Linking durch Limes, was im Programm integriert ist, variieren. Im gleichen Verzeichnis entsteht dann eine Datei **LinkedData.nt**. Diese ist manuell dem TripleStore hinzuzufügen.

6. Weitere Informationen

Falls weitere Fragen oder Probleme auftauchen, verweisen wir gerne auf die in 3. genannten Homepage, wo auch im Falle von speziellen Problemen Kontaktdaten zu finden sind.