

# Vorprojekt

Universität Leipzig – Softwarepraktikum 2014

## Semantic Chess

Jonas Herrig

Hanno Krümpelmann

Nathalie Bargenda

Duc Hieu Nguyen

Stefan Süßmeier

Lisa Höncke

## Inhaltsverzeichnis

Einleitung.....	2
Struktur.....	2
Webinterface.....	3
Vorgefertigte SPARQL-Anfrage.....	3
Virtuoso Triple-Store.....	3
Zusammenfassung und Ausblick.....	3

## Einleitung

Das Projekt „Semantic Chess“ soll dem User ermöglichen durch ein Webinterface einen einfachen und schnellen Zugriff auf eine Datenbank mit Schachpartien zu erlangen. Die Eingabe erfolgt dabei durch Sätze in natürlicher Sprache, die von einem Algorithmus ausgewertet werden und in eine Anfrage an die Datenbank umgewandelt werden soll.

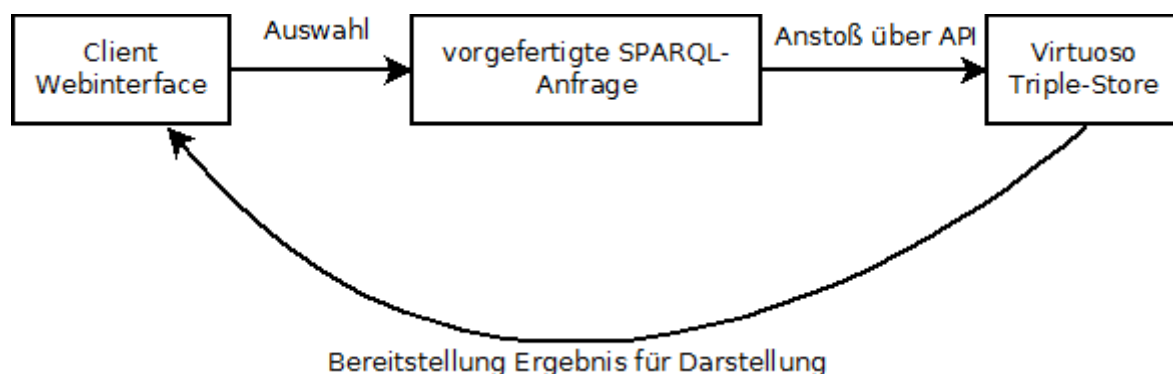
## Struktur

Dabei soll das Vorprojekt die grobe Struktur des vollwertigen Programms bereits umsetzen. Das fertige Programm soll auf der Clientseite eine Weboberfläche für eine Suche in der RDF-Schachdatenbank bieten. Die Suche sowie die Datenbank sollen komplett serverseitig implementiert sein. Das bedeutet, dass auf Seite des Clients keinerlei Funktionalität eingebettet ist – mit Ausnahme von Ein- und Ausgabeelementen.

Den meisten Aufwand wird aus unserer Sicht die Programmierung eines effizienten Suchalgorithmus sowie die semantische Erfassung der Anfrage in natürlicher Sprache verursachen. Aus diesem Grund werden diese Elemente beim Vorprojekt noch nicht umgesetzt.

Darum sieht unser Vorprojekt folgende vereinfachte Planung vor:

Es wird auf Serverseite ein fertiger Triple-Store aufgesetzt werden. Dabei kommt Virtuoso zum Einsatz. Dieser Triple-Store bietet über eine API bereits die Möglichkeit, SPARQL-Anfragen durchzuführen. Clients erhalten die Möglichkeit über ein einfaches Webinterface eine Reihe vorgefertigter Suchanfragen anzustoßen. Der User soll aus einer halndvoll Beispielen auswählen können, die in der Struktur der SPAQRL-Anfrage diverse Use-Cases abbilden. Die Suchanfragen werden mittels ebenfalls vorgefertigter SPARQL-Anfragen durch den Triple-Store umgesetzt und das Ergebnis an den Client zur Ausgabe gesendet.



## Webinterface

Der User soll hier mit einem herkömmlichem Webbrowser auf einer einfachen Oberfläche eine Auswahl aus verschiedenen Anfragen treffen können (ca. 3-5 Optionen). In einem separaten unteren Teil der Seite findet sich Raum für die Darstellung des Ergebnisses. Das Ergebnis wird in Form einer Tabelle dargestellt.

## Vorgefertigte SPARQL-Anfrage

Die zu den Anfragen gehörenden SPARQL-Anfragen werden derart gestaltet sein, dass möglichst eine breite Auswahl an SPARQL-Syntax verwendet wird. Idealerweise bilden diese Beispielanfragen strukturell bereits alle denkbaren sinnvollen Anfragen an die Schach-Datenbank ab.

## Virtuoso Triple-Store

Dieser Programmteil steht bereits fertig implementiert zur Verfügung. Nach einem ordnungsgemäßen Setup des Triple-Stores muss vor allem sicher gestellt werden, dass die API von Virtuoso korrekt in den restlichen Programmablauf eingearbeitet wird. Dies gilt sowohl für die korrekte Übermittlung der Anfragen, als auch für das Auslesen der Ergebnisse.

## Zusammenfassung und Ausblick

Mit diesem Vorprojekt werden bereits die meisten Gebiete der (Java-)Programmierung, welche für die Umsetzung des gesamten Software-Projektes wichtig sind, angeschnitten. Zum einen müssen sich die Teammitglieder vertieft mit SPARQL-Syntax und Virtuoso-API befassen, zum anderen muss bereits die Bereitstellung einer Weboberfläche realisiert werden.

Somit wird nach gelungenem Vorprojekt bereits ein gutes Fundament aus Grund-Klassen und User-Interface vorhanden sein, welches sich dann erweitern lässt. Es kann überprüft werden, ob SPARQL-Anfragen sinnvoll ausgewertet werden im Sinne der angestrebten Funktionalität. Außerdem kann anhand des Vorprojekts die Laufzeit für eine simple Anfrage bestimmt werden. Da im gesamten Projekt die Laufzeit ein wichtiges Requirement ist, können hier schon Rahmenbedingungen untersucht werden.

Des Weiteren sind dank der Webinterface-Lösung auch einige weitere Tests denkbar, wie etwa das Verhalten beim gleichzeitigen Zugriff mehrerer User auf den Triple-Store oder die Verwendung verschiedener Betriebssystem- bzw. Webbrowserversionen.