Group 26: SPARQL and RDF Mining

25/10/2016 **Anirudh Pillai**, Aksel Cakmak and Xiaofeng Fu

Overview

We have established tools to communicate with the client and decided on a time for weekly meetings. We have also learnt more about RDF and used the Stardog GUI and the REST API.

Meetings Summary

Meeting 4 **Team** Meeting (13/10/2016)
We set planned out tasks for the next week. We decided on:

- (Anirudh) Planning development of the R&D Logging Tool
- (Aksel) Create an initial version of the requirements for our project.
- (Allen) Research React and find other components for the stack we will be using.
- (Everyone) Learn about RDF

Meeting 5 **Client** Meeting (14/10/2016)
We spoke with Steffen Stadtmueller who gave us access to Stardog with some example data. Stardog is an enterprise data unification platform. We used the Stardog GUI to explore the example data.

Meeting 7 **Client** Meeting (19/10/2016)
We spoke to Daniel Ewert and found out that they were going to provide us with a running instance of the API so that we won't have to build it locally. They also said that the first draft of our requirements would look very much like the Sta srdog GUI. Additionally they would also want the ability to selectubsets of properties or individuals and the ability to provide a direct link pointing to the selected data.
We also decided to use Gitlab for general communication

Meeting 8 **Team** Meeting (24/10/2016)
The client had provided us with a Stardog with example data and had also hosted the API remotely. We got together to try to access these and to understand how the API works. We also discussed how we were progressing on the tasks we assigned during the last team meeting.

Meeting 9 **Team** Meeting (27/10/2016)
We added the client to our Slack channel. We also started adding tasks to the agile board on Gitlab so that the client knows what we are working on. We have decided on using React for building the frontend.

We also completed the following tasks:

- (Anirudh) Seting up initial infrastructure for the R&D Logging Tool. The homepage and most reusable components have be built.
- (Anirudh) Complete Bi-weekly report
- (Aksel) Initial version of the requirements

Tasks Completed

- Established a good relationship with the client and decided on weekly meetings.
- Refined requirements
- Looked into tools and frameworks for our stack.
- Understood our roles in the team and accomplished tasks accordingly
- Set up an initial version of the R&D Logging Tool

Preliminary MoSCoW analysis

Functional Requirements	Importance
The GUI shall list all classes and properties of the Database.	M
For each class, the GUI shall list all the nodes that belong to this class.	M
For each property, the GUI shall list all the nodes that use that property.	M
For each node in the database, the GUI shall list all the classes the node belongs to and all the properties this node uses.	M
The GUI shall enable the selection of several nodes/classes/properties at the same time.	S
Non-Functional Requirements	Importance
The GUI shall be user friendly.	M
The GUI shall look nice.	S

Plan for next two weeks

The client has provided us with a Stardog Database and a running instance of the REST API. We plan to try writing some scripts to learn how the API works.
We also plan to complete the entire infrastructure of the logging tool including all the pages. We also aim to add content to the initial few pages.
We are also looking into tools and frameworks like Mocha, Chai and Sinon to help us in testing. Additionally, we will also research about application architectures like Redux and Flux which we can use to structure our project.

Individual Sections

Anirudh Pillai

I've learnt more about RDF and also tried setting up some projects in React and the stack that we're deciding on. I've also started creating the R&D Logging Tool for our team and set up a repo for the team to collaborate on it. I've completed the basic structure of the tool and now we only need to add content to it. I've also figured out how to host it on the UCL server.
I've also started to learn how the Stardog GUI and REST API work.

Aksel Cakmak

I created the requirements of the first draft of our project. I divided them by functionality, and assigned an importance to each (following the MoSCoW method).
In order to understand the nature of our project, I spent some time on graph based technologies. More specifically, I used Neo4j, which gives you a way to manipulate graphs through commands inspired by SPARQL. It also has a nice GUI to visualize the data, something that might be inspiring for the future.

Xiaofeng Fu

Last week I still focused on learning about React and i have learnt some about RDF. Moreover, I have meet with our client and know more about our requirements and they said stardog website is our example. And I have joint the github and I will improve the website in next week.