Briefing
Knowledge Engine Project
CSCSI97 Software Design: Principles, Models, Patterns
Philip Lin

Design Document

The design document was very helpful in outlining all of the classes and associated methods of the program. It was particularly helpful in visualizing the connections between the classes, and the abstractions of each class. The design document made it easy to build a basic skeleton of each class before getting into the actual coding and algorithm creation. The descriptions of the methods were important to follow when writing out each of the methods.

It was also very helpful to see the expected output of the program. This made the goals of the program clear and helped to visualize how the queries and the triples were related.

Improvements

The descriptions of the KnowledgeGraph, Importer, and QueryEngine classes could have been improved by defining the roles of each class a little clearer. There seemed to be some overlap between the roles of each class, and so how I interpreted the abstraction of the classes would likely be different to how someone else might interpret it.

It might have been helpful to suggest a specific way to read and extract the Triples and queries from the input files, if there was a favored or most efficient implementation. I used a Scanner with a delimiter and regular expressions to obtain the input that I needed, but there could be more efficient or cleaner ways to achieve the same result. Formatting could have been specified for the output since the identifiers were case insensitive.

Exceptions could have been clearly defined in the document as well so that the types of exceptions that were necessary to deal with would have been clear. Additionally, the result of exceptions could also have been explicitly stated as well.

Changes

I modified the Importer class so that instead of creating and handling Triples, the Importer was only responsible for reading the input file and keeping track of identifiers as Strings. I left the creation of Node, Predicate, and Triple objects to the importTriples() method of the KnowledgeGraph. I wanted to create the actual Node, Predicate, and Triple instances using the class directly associated with the Maps that containes those instances. The functionality of the program was not affected

because I only shifted the responsibility of creating object instances from the Importer to the KnowledgeGraph.

Queries were also kept as Strings, and the queryEngine was responsible only for dealing with query Strings extracted from the input file. In the KnowledgeGraph, query Strings obtained by the queryEngine were used to access the Sets of Triples associated with each query, which was consistent with the way Nodes, Predicates, and Triples were accessed using the getters.

Finally, I added a separate method to print the results of the queries, instead of printing them from directly from the executeQueryFile. The executeQuery method had print commands, but in order to ensure that any queryEngineExceptions were caught before any results could be printed, I didn't use print commands in the executeQueryFile method. The executeQueryFile method creates complete List of queries to execute from the input file if there are no exceptions, and this list can be used in the printResult method to achieve the desired output.