Automatic plug-in workflow construction

Luka Bradesko, Cyc Europe

The thesis will be based on the LarKC platform, particularly on the part which is responsible for smartly connecting and using the plug-ins, to form useful workflows. The result of the thesis will be a Decider plug-in, which is able to construct the most effective and useful workflow, out of the available plug-ins and based on the input SPARQL query.

This is done by using the semantic annotations of the plug-ins, propagating the knowledge in the Platform's plug-in registry and using some sort of reasoning software to decide the ways in which plug-ins can be connected. This can be at the end extended to the point that decider plug-in is able to monitor and switch between the plug-ins on the fly, improving the overall performance.

Data about the plug-ins will be available to the software via the ontology with which the plug-in is annotated. Currently the vocabulary of the ontology is very poor and also the connections between the constants are rare, which makes it obvious that it needs to be improved, especially with the important parameters which are most relevant for making decisions which plug-in to use for what job and provide support for them inside the ontology. These parameters are mostly QoS parameters, together with predicates describing the job the plug-in can do.

The communication between the plug-ins is handled by the platform, but currently data passing consists of just plain triples. There is no way for the computer to tell, what is that it is passing around. In order for the Decider to be able to decide more efficiently which plug-in to use, there should be some way of annotating also the data, not only plug-ins. This implies creating additional ontology which will be used as base when annotating the data.

Also the data passing possibilities which are currently limited to only linear workflows have to be updated to support splitting and joining the data flows, which will make it possible to parallelize the workflow.

The Plug-ins are communicating with the platform via the API, which will have to be improved in order to support all the other features we will introduce while developing. Among other technical improvements, also the API connecting decider and internal platform reasoner (Cyc based) should be defined.

While the LarKC platform already has quite a large number of plug-in available from the community it is still in its early stage of the development and the plug-ins are not always properly annotated. Because of this, we need to generate the test data, based on real plug-ins and also on possible future plug-ins, to show the usability and provide the evaluation for the efficiency and usability of the Decider.