CSc 8630

Project notes

1) (Critical) Identify the overall question to be answered? Is it effectively computable (in other words can you answer it)? Does it involve informatics? Is it worth doing and why?

2) (Critical) Define the steps needed to answer the question. Develop a description of what each step requires and what each step will produce.

3) (Critical) Identify existing software that can be used or adapted as components to solve each step. Most steps will have existing components that can be adapted to work, and can be implemented by using a python/xml-message wrapping. **Web-scraping will not work and will not be accepted.** Web-scraping, in other words wrapping an existing website with programs that mimic user input and parse the output, is problematic and fragile in the best cases. There is no guaranty that the website will be responsive or stable. There is also an ethical issue when you use someone else's computer resources.

4) Design an approach to control the work-flow. The approach will depend on the manner in which the components interact. It can be as simple as each component invokes the next one, but it can also be more complicated – for example using XMLRPC to invoke local servers to handle individual tasks. It is most likely not necessary to use heavy-weight systems like COBRA.

5) We would like to have a web interface and some mechanism for returning the results. It is unlikely that the calculation will proceed in real-time so an email response is probably best.

6) (Critical) Instantiate use cases or acceptance tests for each component. Do not expect to finish components in a sequential manner. It will be necessary to develop them in parallel. The use cases establish a contract between the developers of each component, which makes this possible.