UNIVERSITY OF Southampton
School of Engineering Sciences

An Intelligent Workflow Construction Support System for MATLAB

UTC for Computational Engineering

Hakki Eres and Andy Keane: CEDG School of Engineering Sciences

Introduction

In the process of intelligent workflow support there are two distinct roles. The first role is the one of the expert who possesses deep knowledge in the domain, in our case design optimisation; the other is the role of the end user, who still needs help and guidance from an automatic and intelligent workflow support system in order to develop complex workflows that require some knowledge in design optimisation.

The overall system architecture of the SIMDAT intelligent workflow construction support system architecture is given in **Fig. 1**, and the distinct boundary between the domain expert (or the knowledge engineer) and the end user is shown with a dotted line. The domain expert uses a publicly available, open-source, and widely used ontology editor called Protégé. This program allows the domain expert to construct expert domain knowledge in a graphical environment.

As the knowledge model grows, it gets more and more difficult to maintain the consistency of the model manually. Therefore, automatic reasoners, are needed to check the consistency of the knowledge model. These reasoners can further be used to infer additional knowledge based on the asserted knowledge model. A snapshot of the engineering design search and optimisation ontology model is given in **Fig. 2**.

When the domain expert is satisfied with the consistency and the usefulness of the knowledge model they can save the model in an OWL file and upload the file to a triple store, in our case Sesame server. Depending on the installation, actual data can be stored in a local database application like MySQL, in memory, or on a local filesystem. The access to knowledge repositories are password protected, and by using a very simple user interface the domain expert uploads the OWL model to the repository. During this process the Sesame server checks the consistency of the model one more time, and warns the domain expert if inconsistencies exist.

MATLAB Interface

We assume that the end user needs to create an optimisation workflow or script in the MATLAB environment, but they have somewhat limited knowledge on existing methods and techniques to apply to their design problem at hand. Therefore, we use MATLAB's Java support along with the Sesame API to implement MATLAB functions to provide the end user an intelligent workflow construction environment. At the simplest level this will involve a basic help system that offers solutions on the details of different algorithms and optimisation techniques, and at a more complex level the user might need to know what would be an appropriate population size for a genetic algorithm, or whether there are any methods for a specific problem that would provide a better convergence. The MATLAB interface consists of some basic functions that perform ontology file uploads, test the status of Sesame triple stores, and more complex functionality to perform various database queries.

SeRQL Queries

When users are sure that the knowledge server is up and running, they can perform various queries on the Sesame triple store by using implemented MATLAB query functions. In **Fig. 3** the user first searches for a comment attached to a specific concept called "Currency," and then they use a wildcard to search for all the comments of all concepts.

For the next example we have created two repositories, the first one only holds the asserted ontology which is not capable of using reasoning techniques, and the second repository can use automatic reasoning techniques to infer additional knowledge on the same ontology. When the user queries the concept "Numerical" from the asserted repository the query returns only one result. However, the inferred repository returns more knowledge to the end user as shown in **Fig. 4**.

In the optimisation ontology generated by the Computational Engineering Design Group there are specific equivalent names that are used to describe the same optimisation algorithm. For these type of concepts the also exists another MATLAB function, called oa_QueryRepositoryForEquivalence, that allows the end user to perform equivalence queries as shown in **Fig. 5**. Note that the inferred repository returns automatically generated knowledge.

Conclusions

An intelligent workflow construction support system has been implemented in order to aid domain experts to construct a useful ontology and to help end users to query and retrieve this knowledge easily by using open source software packages. This work demonstrates that by integrating only open source software, it is possible to implement a useful environment that allows ontology creation and reasoning, as well as knowledge maintenance and querying capabilities within a workflow construction environment.

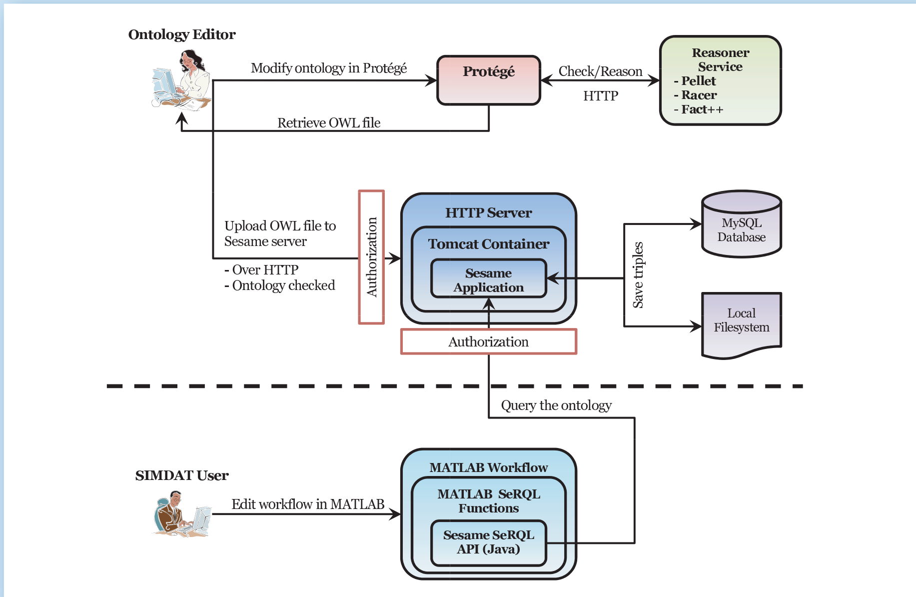

Fig.1 Overall system architecture.

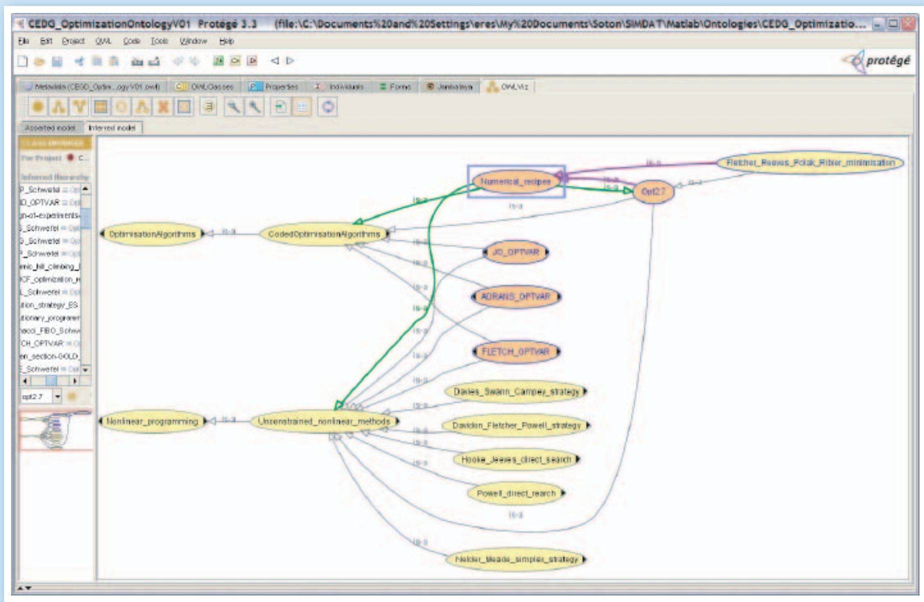

Fig.2 Inferred model showing the numerical recipes method as a direct subclass of unconstrained nonlinear methods and coded optimisation algorithms.

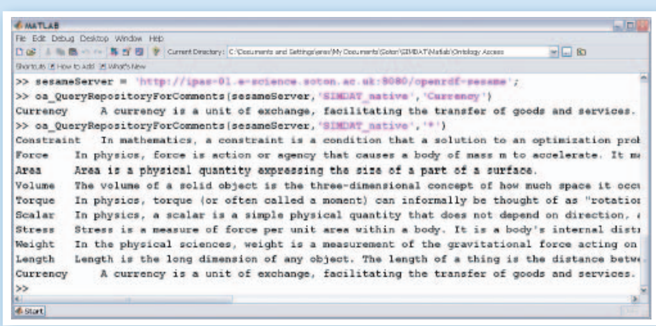

Fig.3 Querying for comments of concepts by using a specific search argument or a wildcard.

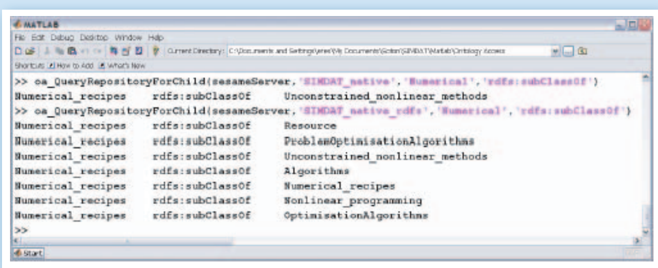

Fig.4 Querying asserted and inferred repositories for the same concept.

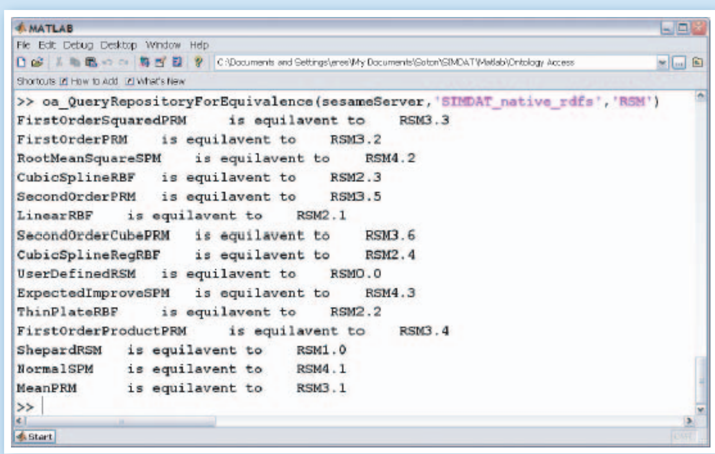

Fig.5 Querying for equivalence from the inferred repository.

Acknowledgements: This work is supported by the SIMDAT project (European Commission contract number IST-2004-511438).

www.soton.ac.uk/ses/research/ced/posters.html | email: Hakki.Eres@soton.ac.uk

Computational Engineering & Design Group, University of Southampton, SO17 1BJ, U.K.