Collaborative Capture and Reuse of Environmental Knowledge

Jochem Liem and Bert Bredeweg
Human Computer Studies Laboratory – University of Amsterdam
Kruislaan 419 (Matrix I), 1098VA Amsterdam, The Netherlands
{jliem,bredeweg}@science.uva.nl

Introduction

Searching for existing models concerning a particular topic in environmental science and assessing its relevance is often experienced as being difficult. Actually reusing part of a model in a meaningful way brings along many more problems, and it therefore rarely done. However, an environment that would support such activities would enhance the collaborative capture of environmental knowledge by relieving researchers of redoing modelling work that has been done before by reusing earlier work.

Within the NaturNet-Redime project we have created a working environment around the qualitative modeling and simulation tool Garp3 (http://www.garp3.org) that supports the searching, assessing, and reuse of knowledge formalized in qualitative models. This paper describes the qualitative model library, the Web Ontology Language (OWL) formalization of the models needed for this library, and copy functionality in Garp3, which are part of this working environment. These features should increase the amount of reused model parts in the environmental modeling community.

Garp3 – QR Workbench

The Garp3 workbench implements a diagrammatic approach to modeling and simulating qualitative models. Modeling in Garp3 starts by creating model ingredient definitions. These definitions include entities, agents, assumptions, configurations, quantities and quantity spaces. Entities, which represent the structural objects in a system, are organized in a sub-type hierarchy. They are defined by their name and their position in the hierarchy. Agents and assumptions are defined in the same way. Agents cause influences from outside of the system, while assumptions are labels that indicate that certain conditions are presumed to be true. Configurations are structural relations between entities that are defined by their name. Quantities represent the features of entities and agents that change during simulation, and are defined by their name and a set of possible quantity spaces. Quantity spaces represent the possible values a magnitude (or derivative) of a quantity can have, and are defined by their name and an ordered set of possible values. Quantity spaces are associated to the quantities of entities or agents.

Next to the model ingredients defined by the modeler, there is also a set of predefined model ingredients. These include causal dependencies (proportionalities and influences), correspondences, the operator relations plus and minus, value assignments, and inequalities. All these model ingredient definitions can be used (instantiated) to create model fragments (MFs) and scenarios. MFs can be seen as composite ingredients that incorporate other ingredients as either conditions or consequences. They are organized in a subtype hierarchy, meaning that a child MF inherits the model ingredients of its parents. Furthermore, a MF can incorporate other MFs as conditional ingredients.

MFs instantiated in another MF are called Imported Model Fragments (IMFs). An example MF incorporating another MF twice is shown in Figure 1.

Scenarios are also composite model ingredients. They describe specific system situations. During simulation, MFs are sought which match on the scenario (i.e. the model ingredients fulfill the conditions of the MF). The consequences of matching MFs are merged with the scenario to create an augmented state from which the next states of behavior can be determined.

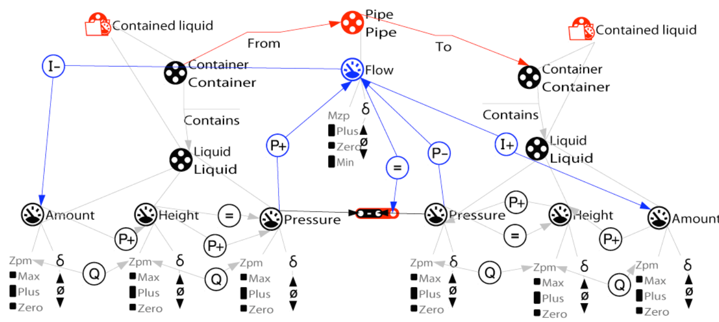

Figure 1: Liquid flow includes 2 Contained Liquid IMFs, a Pipe, and the configurations as conditions, and flow, its calculation and causal relations as consequences.

Model Repository

To reuse models of others, modelers have to be able to share their work and access work of others. This is solved by allowing models to be uploaded to and downloaded from a central online model repository. However, the number of models in the repository can potentially become large, which means that modelers need to be supported by search functionality to find reusable models.

Typically, a modeler will want to search for models which contain a certain entity or quantity (e.g. a model which contains both an entity *population* and a quantity *size*). Normal search engines search for keywords in text and are unable to interpret the explicit knowledge representation in qualitative models. So the search engine is unable to distinguish between different types of model ingredients, or between domain specific and domain independent knowledge (i.e. the QR vocabulary and the knowledge formalized by the modeler). This hampers the search engine's ability to find relevant models. A search solution should make use of the explicit knowledge representation in qualitative models to allow modelers to focus their search using the QR vocabulary.

The Semantic Web initiative proposes that "semantic search" becomes possible by making content machine-accessible [1]. The Web Ontology Language (OWL) is a description-logic based knowledge representation language, which is represented in RDF/XML, and is being developed as part of the Semantic Web initiative. It has become the de-facto standard for the sharing of knowledge on the web in the form of *ontologies*. By formalizing qualitative models as OWL ontologies, the models become interpretable by OWL search engines, and searching for models in which certain model ingredients or certain structures are used becomes possible. Additionally, the formalization of models in OWL opens up the possibility for other QR-tool developers to implement functionality to import these files. This could potentially make models accessible to communities using different QR tools.

A qualitative model repository was implemented (see Figure 2) as a webpage that allows modelers to share their own models as OWL files, and search and download models of others. The search functionality shows the model ingredient definitions of all the models. Selecting a definition reduces the list of matching models, allowing the modeler to iteratively refine the list of potentially useful models. An initial assessment of the usefulness of the models can be done by reading their metadata. Chosen models can be downloaded and investigated in Garp3.

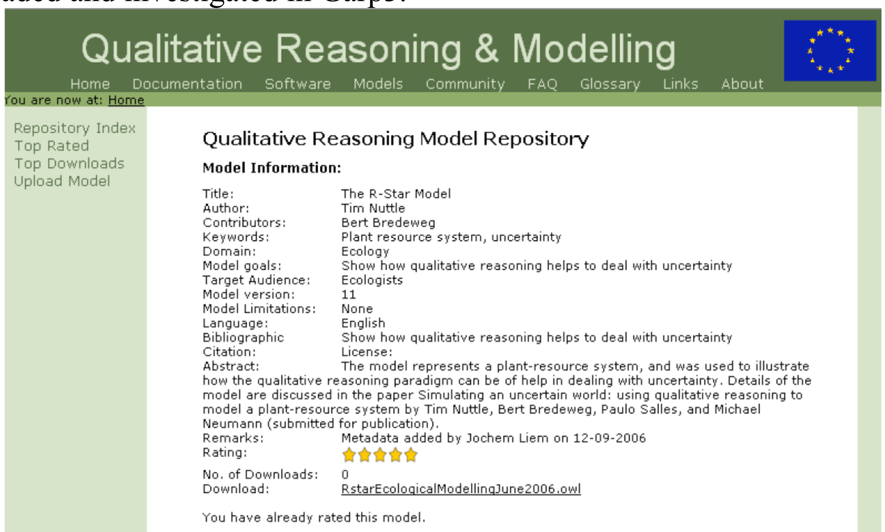

Figure 2: The Qualitative Model Repository.

Reuse through Copy Functionality

The Garp3 workbench allows users to open multiple models at the same time, and copy model ingredient definitions, model fragments and scenarios from one model to the other. There are several issues that have been addressed to make this possible. Firstly, model parts often depend on other model parts without which they are incomplete. Garp3 ensures the completeness of the copied knowledge by creating a minimal self-contained sub-model each time a model part is copied. Secondly, syntactical correctness of models is maintained automatically. Since Garp3 is a diagrammatic tool that automatically ensures syntactical correctness, this correctness is maintained when copying model parts by creating the model parts in the next model as if they would be created from scratch. Thirdly, no redundant knowledge is added when model parts are copied, as similar model parts already existing in a target model are reused. Fourthly, the semantics of the model parts is preserved as much as possible, putting entities, agents and assumptions as close below their parents in the sub-type hierarchy as possible. Finally, existing model parts in a target model are not altered. This assures that alterations to existing simulations of the model can be easily restored by deleting the new model parts. Copied model ingredients with clashing names are created with the suffix '(copy)' behind them. The copy functionality makes it easier to reuse model parts from existing models.

Acknowledgements

This work is co-funded by the European Commission within the Sixth Framework Programme (2002-2006), project NaturNet-Redime (http://www.naturnet.org), number 004074.