Supporting the Construction of Qualitative Knowledge models

Bessa Machado, V.
Conclusions and Discussion

This thesis started from the idea that modelling and more specifically qualitative modelling is a central skill in scientific reasoning. Enabling and encouraging the creation of domain theories, which can be instantiated to model specific situations, should help one to understand the broad applicability of scientific principles and processes. Qualitative modelling provides formalisms for expressing intuitive, causal models and the reasoning techniques needed to generate predictions and explanations from them, thereby explicitly rendering the consequences of the ideas that someone has. Qualitative reasoning formalisms provide the expressive power needed to capture the intuitive, causal notions of many human mental models.

While the representational ideas of qualitative modelling are ideal for science education, the formalisms typically used to express them are not directly suitable for learners because their use requires expert level programming. Making these formalisms available through graphical means is, we believe, the missing piece of the puzzle for making the full potential of qualitative reasoning accessible to a broader audience. Models provide a means to externalise thought. Graphical representations help reduce working memory load, allowing students to work through more complex problems. Such external representations also help them present their ideas to others for discussion and collaboration.

We set out to accomplish the ambitious goal of defining a domain-independent model-building methodology as well as designing and implementing a corresponding model-building workbench. The workbench should be usable by a wide range of users, active in different kinds of science teaching curricula. Three main research questions followed from this and guided the research carried out in the PhD project. Firstly, what are the tasks involved in qualitative model building? Secondly, how will the model-building environment be visualised on screen, particularly concerning the model contents? Thirdly, what kind of support should be provided for users while constructing a model?

A series of activities were engaged with the aim of defining the optimal model-building methodology and implementing a user-friendly, broadly applicable software package for model construction. Below the important conclusions of this enterprise are enumerated

7.1 Main Results

Based on the existing qualitative reasoning engine GARP a rational task analysis of model construction was carried out. It turned out that a qualitative model-building in general
consists of seven main activities. For each of these a detailed task-analysis was carried out, specifying the tasks that should be performed, as well as by whom (user or system). In doing so, the requirements for the design of a modelling workbench were specified in terms of the functionality that such a software tool should have.

To design the graphical user interface of the workbench a set of knowledge-visualisation principles was developed. After analysing the typical characteristics of QR-based knowledge models in terms of the ingredients that such models consist of, the following main guidelines were defined. Firstly, icons should be used for representing the information about the model-type of an ingredient, because this kind of information is fixed as it is fully determined by the ontology underlying the reasoning engine. Secondly, the visual representation for ingredients that convey relationships between other ingredients is best implemented using a line, connecting the related items, with either an icon or a text label. The choice between the latter two depends on whether the relationship is user defined (text label) or pre-defined by the modelling language (icon). Thirdly, in order to preserve consistency, icons representing model-types should always be displayed whenever a class of that model-type is shown. Finally, relations that do not exist explicitly in the model-building language (such as 'belongs to') can be represented as thin lines between the model ingredients (without a uniquely defining icon attached to it). More in general, each builder has one default relation for which a defining icon can be omitted.

In addition to the above, a more technical design was specified for the model-building environment in terms of builders, tools, graphs, vertices and arcs. The builders are the workspaces for handling a set of model ingredients. The model in each workspace is represented as a graph that consists of vertices (ingredients) and arcs (ingredients that relate two other model ingredients). Following the task analysis seven builders were designed. In order to interact with the builders, tools are defined to create, modify, and delete ingredients within the builders. Next to this basic functionality each builder has dedicated functionality. The behaviour and functionality of the tools is driven by the specific interaction requirements within each of the seven builders.

HOMER is a fully working prototype model-building tool, following the requirements discussed above. One of the goals of the experiment with HOMER was to validate the task analysis. The experimental results were satisfactory from this point of view except for the task of editing quantity spaces. This task, contrary to the task analysis, was not seen by subjects as an independent task but was mainly performed within the context of creation quantities.

A second goal of the experiment with HOMER was to assess the usability of the tool. The Usability Heuristics [83] were used for this purpose. Two heuristics were violated significantly more then others, namely 'Visibility of Systems Status' and 'Error Prevention'. These are a typical problem for complex workbenches. However, an important conclusion following the usability analysis was that, although there were usability problems, these did not prevent users from realising their model-building goals.

The third goal of the experiment with HOMER was to establish the typical conceptual problems that users may have when building qualitative models. The experiment was successful in this respect as it highlighted a range of specific problems. These problems were analysed and categorised into four main classes.
7.1. **MAIN RESULTS**

- **Model scope.** Determining which features of the original system to include in the model. Such problems may for instance be to finding the relevant quantities of a system.

- **Model structure.** Determining what to put where in the model. For instance, the issue of deciding on the type and number of model-fragments that are needed. The notion of re-use is important in this respect, because it provides guidelines for thinking about how to structure the model. For instance, it is possible to capture all the details of the U-tube system in a single model-fragment. But such a model cannot be (re-)used for reasoning about the behaviour of containers, substances and flows in general.

- **Model-building concepts.** Understanding the model building concepts as provided by the tool. For instance, the difference between attributes and quantities, the meaning of an influence, or the difference between generic and instance knowledge. Model builders need to understand the qualitative ontology as used by the tool. They must learn to use it in order to acquire more advanced insights about the system behaviour they are trying to model.

- **Model representation.** This is related to the 'model building concepts' category, but now it refers to the actual representation of an idea using the model-building ontology. The user wants to articulate something but does not know how to technically formulate that with the options provided by the environment.

Within these four categories, problems caused by failure to understand the model building concepts were the most frequent ones. This result may be explained as follows. Firstly, this category reflects the understanding the user has of the role of each ontological primitive. These may sometimes be unclear for novices, especially when regarding the use of the different kinds of dependencies. Secondly, the environment forces the user to make a clear distinction between generic and instantiated knowledge. It was frequently noticed that users had difficulties in differentiating the two.

To establish an improved second prototype, particularly addressing the conceptual model-building problems that user have, a study was conducted analysing other existing software systems that may have features relevant to HOMER. This resulted in a set of 'lessons learned' categorising and focussing the different kinds of support that the new prototype should be able to deliver.

- **Domain Independence** brings about a number of consequences which make the provision of support more demanding than in the case of domain dependent systems. For instance, the graphical language is an extra feature to be learned and additional support must thus be provided.

- **Qualitative modelling** is usually not part of standard curricula and therefore support related to the QR ontology and mechanisms have to be made available.
CHAPTER 7. CONCLUSIONS AND DISCUSSION

- The simulation results are per se a form of feedback and induce users to reflect. Therefore, being able to run a simulation and analyse its results in the same environment is an essential feature of a well-built model building environment. Furthermore, having multiple forms of presenting the simulation results is desirable since this facilitates the interpretation of those results.

- Due to the impractically high engineering efforts required to elicit domain dependent knowledge about problems in users' understanding of a domain, but also due to having opted for a constructivist approach, we do not intend to implement a student model, as done in traditional ITS systems. However, we do provide tailored feedback based on the modelling results and knowledge about required model structures.

- Pedagogical agents represent a new learning paradigm. We have thus proposed a set of agents that can guide users in the process of building models. The use of different characters to handle different aspects of support has been suggested. We are herewith taking advantage of the personification and modularity features of agents.

Based on the lessons learned we developed a framework for supporting the QR modelling process. It provides support related to conceptual knowledge, including the model-building ontology. It also provides tailored feedback addressing the individual needs of a user.

On the basis of the guidelines and requirements as well as the lessons learned from the experiment and the literature, a second - improved - prototype was built: MOBUM. MOBUM is a fully working WEB-based model-building workbench. It was designed and implemented to address the usability issues and conceptual problems discovered while working with HOMER. MOBUM integrates support for the three main phases in modelling: model creation, behaviour prediction, and simulation inspection. For the model creation part, MOBUM consists of construction components and specification components. Construction components are the tools that provide the modeller with the basic model-building constructs for building qualitative models. They are the Structure Builder, Quantity Builder, Quantity Space Builder, Model Fragment Builder and Scenario Builder. In addition, specification components are tools that support the model-building process by allowing articulation of knowledge in a different and less restricted formalism. These are 1) a tool that facilitates sketching and 2) a tool that facilitates causal model creation. For the phases of behaviour prediction and simulation inspection, MOBUM includes the "GarpApplet" tool. It enables the user to directly run the model and provides means to inspect the results, thereby making the modelling process a meaningful and complete experience.

Searching for help is more efficient when the support system is based on modular processes. We opted for an agent-based approach to providing support in which each agent is specialised in some specific task and together with the other agents collectively contributes to the achievement of a global objective. Agents thus have scope, provide context-sensitive help, and are personified according to the type of support they provide.
Four categories of support were defined: static (pre-defined), dynamic (tailored to user needs), local (within a builder), and global (within a builder, but taking the whole model into account).

Two studies, one with novices and one with experts, were performed to assess the usefulness of the support module and the usability of the user interface. The results of our studies were encouraging. Most of the problems that novices encountered were or could have been solved by consulting the agents, which reinforces the idea that MOBUM indeed supports the model building process. Within the experiment involving experts, the absolute number of errors, when compared to the HOMER experiment decreased drastically (67 in HOMER and only 10 in MOBUM). Furthermore, all problems encountered by the participants could be solved by consulting the agents. In fact, the dynamic help was well appreciated by the participants. As one of the participants put it: "the agents helped me keep on track".

A third study was performed especially to compare the two model-building environments, MOBUM and HOMER. The goal was to evaluate whether the new prototype was more effective and if it would be more appreciated by the users. Based on the results we cannot prove that MOBUM is more effective, because the variation in the models creating during the experiment was too large to make a reasonable comparison. However, it is safe to conclude that the improved usability and (intelligent) support influences the evaluation of the tool: subjects evaluated MOBUM significantly more positive.

Given the GARP reasoning framework, the design of the model-building workbench proceeds via the clarification of the ontology involved, a detailed task analysis, a consistent set of visualisation principles, and the implementation and evaluation of a number of prototypes. In addition methods for providing support in a principled way, were developed. This methodology enables the construction of complex model-building tools that are usable and adequately support the model building process.

7.2 Future Developments

The evaluation of MOBUM in practical situations brought about a number of ideas for future developments which will be described next.

Instead of just supporting the building of models from scratch, the tool could be extended in order to allow the execution of intermediate modelling tasks. For instance, having the facility of running a simulation on the isolated causal model as specified in the causal model-builder would be an interesting additional feature.

An additional improvement for the model-building environment would be to provide the means allowing diagrammatic sketching of ideas and conceptual knowledge and have these automatically transformed into a valid simulation model. The Swan SketchPad was the first step towards accomplishing this goal. However, more research is needed on how to fully implement this idea. This would also involve advanced means for model debugging, which would alert the modeller to potential logical errors or omissions.

Collaboration between several modellers can be either simultaneous or sequential. In the latter case, modellers adapt (re-use) a previously developed model to a new purpose.
In this case, the modelling coach should implement the important function of explaining to the new user how the previously created model works. Dedicated communication tools that allow collaborators to exchange information about specific problem areas in an efficient way also need to be developed. Such communication tools would also support the important task of model documentation, which will become increasingly important in sequential collaboration.

Besides developing an environment that supports qualitative modelling, in order for this environment to be used in classrooms, we need to develop curricula that are integrally based on QR models for a variety of subjects. These curricula would include, for instance, interactive models, practical exercises, and assignments. QR models, exercises, and assignments can be developed that capitalise on the potential of QR to facilitate causal reasoning by allowing the students to interact with the model. Development of lectures that incorporate QR representations of important domain-concepts needs also to be supported. With increasing development of QR models, assignments, exercises, and lectures for a given domain, the integration of QR approaches with traditional teaching approaches needs to be supported. Such integration will draw on the ability of QR to enhance conceptual understanding while also maintaining connections with traditional approaches that are essential for students to understand foundational materials for their domains.

Having constructed a model building environment to support modelling with the assumption that modelling is important to acquire conceptual knowledge, we need then to validate this hypothesis. Until now the evaluation process has been mainly centred around the usability of the tool. However, evaluation must be performed in specific educational settings involving science teaching, for instance in a high school classroom. In [110] a study is presented which provides guidelines for research on the development and use of model-building tools in educational settings. However it was a preliminary study and additional research is needed into this subject.

Recently, studies have been presented that use a related approach and provide learners with tools to construct diagrams that represent causal explanations of system behaviour, notably 'Betty's Brain' [13] and VMODEL [17]. The primitives provided by these tools for knowledge creation are based on qualitative formalisms, particularly on the Qualitative Process Theory [43]. However those tools particularly focus on learners assembling causal behaviour diagrams. Our work further develops this idea of 'learning by building models' by having learners construct full qualitative models, run simulations, and thus support learners in a wider range of abilities. A thorough evaluation of the learning outcomes still needs to be done.

The idea of model-building communities introduces several interesting problems on our research agenda. While working on this agenda, more and more elements are being put in place, alleviating part of the shortcomings of today. Therefore, we believe that interactive model-building environments based on a domain-independent modelling ontology will prove very useful for the communication of knowledge in the future.