Towards support in building qualitative knowledge models

Vânia Bessa Machado, Roland Groen and Bert Bredeweg
Human Computer Studies, Faculty of Science, University of Amsterdam,
Kruislaan 419 (Matrix 1), 1098 VA Amsterdam, The Netherlands
vbessama, bredeweg@science.uva.nl

Abstract

Qualitative Reasoning (QR) formalisms provide ontological primitives for capturing *conceptual* knowledge. Recently QR-based diagrammatic tools are being developed to support learners in creating concept maps as means to acquire such knowledge. However, QR formalisms are complex which hampers their usability. While other approaches address this by simplifying the formalism they use, we seek the solution in providing a set of agents that can support the learner. Based on previously reported study on using QR modeling tools, we have developed a multi-agent approach to support the QR modeling process. The agents provide different kind of help, such as general information on the formalism and tailored feedback addressing the individual needs of a learner. Agents thus have scope, provide context-sensitive help, and are personified according to the type of support they provide. An evaluation study shows that the help-system is well accepted by learners.

1 Introduction

Conceptual analysis of systems and their behaviour is a central skill in scientific reasoning. Enabling and encouraging the creation of domain theories, which can be instantiated to specific situations, helps learners to understand the broad applicability of scientific principles and processes. The research area Qualitative Reasoning (QR) provides means that can aid this kind of learning. QR formalisms provide a way to express *conceptual* knowledge such as system structure, causality, the start and finish of processes, the assumptions and conditions under which facts are true, qualitative distinct behaviours, etc. Models provide formal means to externalise thought on such conceptual notions. Particularly the idea of having learners learn by *building* qualitative knowledge models enables them to formulate their own ideas, test them by simulation, and revise them were needed [Collins, 1996; Reif and Larkin, 1991]. These are important scientific skills for learners to acquire.

QR formalisms are complex and therefore not always easy to use in educational settings. Recently tools are being developed that take a graphical approach to having learners build qualitative models [Bredeweg and Forbus, 2003]. Graphical representations help reduce working memory load, allowing students to work through more complex problems. Such external representations also help them to present their ideas to others for discussion and collaboration. This closely relates to the idea of using concept maps [Novak and Gowin, 1984]. The main difference being the rich and detailed semantics used, which are based on QR formalisms. To further enhance usability, approaches such as Betty's Brain [Biswas *et al.*, 2001] and Vmodel [Forbus *et al.*, 2001] reduce the amount of primitives available in the model-building tool. Although this is effective, it has the obvious drawback of not using the full potential of QR and the means it provides to articulate conceptual knowledge. In our approach we want to preserve the full expressiveness of the QR formalism. To enable usability we have develop support tools that aid learners in understanding the representational primitives (which we regard as an important learning goal by itself) and to articulate and reflect on their thoughts.

This paper discusses the multi-agent help system that we have developed for the domain-independent model-building environment MOBUM [Bessa-Machado, 2004]. It builds on previous work [Bessa-Machado, 2004; Bessa-Machado and Bredeweg, 2003] in which we used the workbench Homer to evaluate the usability of a diagrammatic representation for qualitative knowledge and the need for additional help, both from a learner perspective. The evaluation of Homer was designed such that we obtained as much problems as possible that learners encountered when working with Homer. Based on the insights gained from this evaluation MOBUM was constructed. MOBUM uses a related but improved diagrammatic presentation, compared to Homer. To further enhance usability, MOBUM was given a multi-agent help system that is capable of providing useful help without maintaining an explicit learner model nor a norm model.

The organisation of this paper is as follows. The following section briefly explains some of the main characteristics of MOBUM. Next, the help system is discussed in more detail. We then discuss the evaluation of MOBUM with learners.

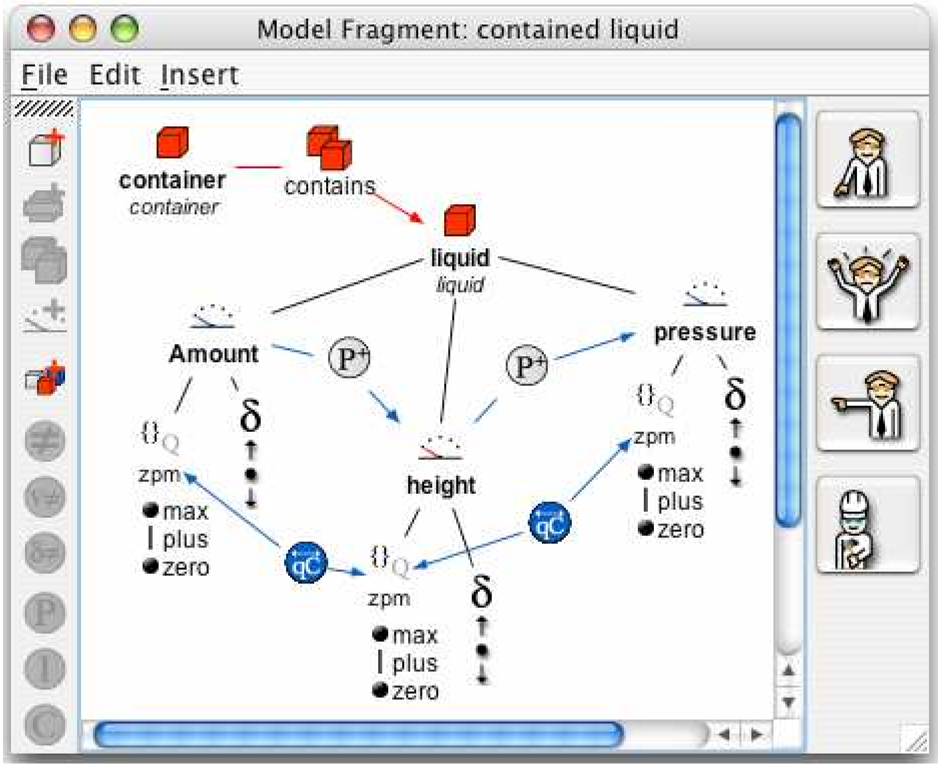


Figure 1: Model fragment of a 'Contained Liquid'

2 MOBUM - a brief overview

MOBUM is workbench for creating and simulating qualitative knowledge models. It is based on the QR formalism described in [Bredeweg, 1992]. The graphical user interface of MOBUM is organized as a set of builders and tools. Builders are interactive windows that support the learner in building specific model ingredients. In the current version of MOBUM there are five builders that support the creation of these model ingredients, namely for: Model fragments, Quantities, Quantity spaces, Entities and Scenarios. Two others builders exist that do not directly add content to the model, but support the learner in exercising his/her understanding of the system being modelled. These addition builders provide means for expressing ideas using drawings (SWAN SketchPad) and causal dependencies (Causal Model Builder). In addition to the builders there is a set of Model Inspection Tools, which allow the learner to run a simulation, to visualise the global simulation results (e.g., state-graph) and to inspect the specific results of the simulation (e.g. the contents of an applied model fragment). After running a simulation, the modeller will get a state-graph and can verify, for instance, how the quantities behaved in the different states, which model fragments have applied, the content of a specific state, and how the transition occurred from one state to another. The diagrammatic representation of model ingredients within the builders follows the guidelines presented in [Bessa-Machado and Bredeweg, 2003]. For example, *Quantities* in the Quantity Builder are organised in a list, because no relation exist between them, while *Entities* are represented as nodes in a graph and the *is-a* relation between the entities are represented as arcs between those nodes. An example of what a learner may produce is shown in Figure 1.

The figure shows the Model Fragment Builder that captures generic knowledge about a container containing a liquid, hence 'contained-liquid'. The single cube-like icons represent objects (*container* and *liquid*). The double cubes represent structural relations between objects (*contains*). There are three quantities: *amount*, *height*, and *pressure*. They are assigned to the *liquid*. Each of these quantities can take on three possible values *zero*, *plus*, *max* and they can be *increasing*, *steady* or *decreasing* (∂) (although in this example no specific values nor derivatives have been assigned). *Amount* has a positive influence on *height*, and *height* on *pressure* (P+), which means that when *amount* increases, so will *height* and *pressure*. These proportionalities (P+) are *directed* causal dependencies. Thus: a change in the *amount* causes the *height* to change and not the other way around. Finally, the quantity spaces for these three quantities fully correspond, which means that they will always have the same value, e.g. all having value *max*. Notice that most of these model ingredients have been created with the other builders, such the Entities, Quantity, and Quantity space Builder. In the Model Fragment Builder these ingredients reused are related. In fact, only the Correspondences (qC) and the Proportionalities (P+) are actually added in the Model Fragment Builder.

3 The Agent-based Help System

The design of the help system is based on the results from the study with HOMER [Bessa-Machado and Bredeweg, 2003]. The help system should be usable for a wide range of learners, active in different kinds of science teaching curricula. It should provide support related to conceptual knowledge, including the model-building ontology, and it also should provide tailored feedback addressing the individual needs of a learner.

Taking a domain independent approach has at least two consequences. Firstly, besides providing support to the learners in acquiring conceptual knowledge, support concerning the graphical language must also be given. As a result of being domain-independent, the icons used in MOBUM are generic and learners will most likely not immediately associate the underlying concepts with their visual representations. Secondly, the use of a learner model, in the traditional sense, is not possible because it would require a domain specific norm model to work from. To cope with this situation, we take a rather different approach compared to traditional ITS systems. Instead of focussing on the domain knowledge that the learner is supposed to acquire, we focus on the processes that are expected to lead to the acquisition of that knowledge. That is, we provide tailored feedback based on knowledge about the model-building process in general and the constraints following from the specific model built by a learner. Another feature of our approach is that the support system takes the form of an advisory system. We do not want to interrupt the learner in order to offer help. The learner is in control and can initiate a support session if needed. Using pedagogical agents is a relative new paradigm. Furthermore, searching for help is more efficient when the support system is based on modular processes. We opted for an agent-based approach 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. Two main categories of support were defined: static (predefined), dynamic (tailored to learner activities).

3.1 Structure of the Help System

Since the applicability of *static* and *dynamic* information is clearly delimited, their availability should also be broken down into discrete stages. Similar to what was done in the work presented in [Shimoda *et al.*, 2002] and in order to stimulate the use of help as well as to unambiguously characterise each type of knowledge support, six agents presented as different characters are used (Table 1).

Each agent has a specific appearance representing the type of support it can provide. Each builder, representing a particular step in the model-building chain, possesses its own implementation of the various agents (e.g., the model fragment builder has four of these agents, see Figure 1). The whole set of agents is thus present at all times but the support provided will depend on the actual model-building context.

3.2 Static Help

The *static help* is implemented as two complementary forms. Firstly, by providing definitions for the terms composing the model-building ontology, and secondly by giving examples on how to use those terms. The static help system is thus able to answer questions such as *'What is an influence?'* and explain *'How to create?'* an influence using the available tools.

To support the learner in solving a problem, static agents use explanatory text, examples and images. The information is displayed inside a dialog box using HTML pages including hyperlinks and cross references. Images are also used for displaying MOBUM GUI parts. Four static agents are included in the design. They are labelled according to their specific utilities: *What is*, which has the task of helping learners on model-building concepts in the actual builder; *How to*, which suggests the order in which modelling steps should be performed and the actions needed to reach a certain goal; *Curriculum planner*, whose goal is to provide information related to specific assignments given to students; and *Global help*, which is knowledgeable about general modelling issues. It also explains the application of all ontological primitives and discusses basic ideas on how to create a model.

3.3 Dynamic Help

The dynamic help provides support relevant to the specific *content* of the model being created. This type of help thus needs to have *assessment* capabilities concerning the prior and actual user production in order to be able to evaluate the progress of the learners. Since this progression is a dynamic process, the contents of the provided help will be changing constantly. The dynamic help continuously analyses the current solution of the learner to the assigned problem and compares the steps taken to reach this point with a selection of generic correct modelling features. Any inconsistencies will be detected and can be reported to the learner so as to instigate the learner to reflect on the actions taken and maybe consider an alternative trajectory. By doing so we try to keep learners on track and to avoid them arriving at incomplete models.

The dynamic help system is designed to provide guidance at two distinct levels: *local* and *global*. The former is concerned with the details of a specific modelling subtask and is usually restricted to a certain builder. The latter, on the other hand, gives a global perspective on the modelling activities of the learner, reuniting the actual status of the full model. This distinction between *local* and *global* knowledge is an important one, since the construction of models will usually be a constant interplay between figuring out the fundamental details of the underlying model ingredients and defining the overall relationships between those ingredients. Two dynamic agents were designed to provide tailored advice and suggestions on both local and global aspects of the model. They have been denominated: *What can I do next?* (local) and *Cross builder help* (global). At the local level, help is generated on the learner's actual model-building activity. The help facility analyses the input of the learner within the active builder and guides the learner by providing a set of

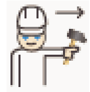
	What is?	How to create?	Curriculum Planner	Global Help	What can I do next?	Cross Builder Help
Standby						
Active						

Table 1: Agents characters in the MOBUM multi-agent help system

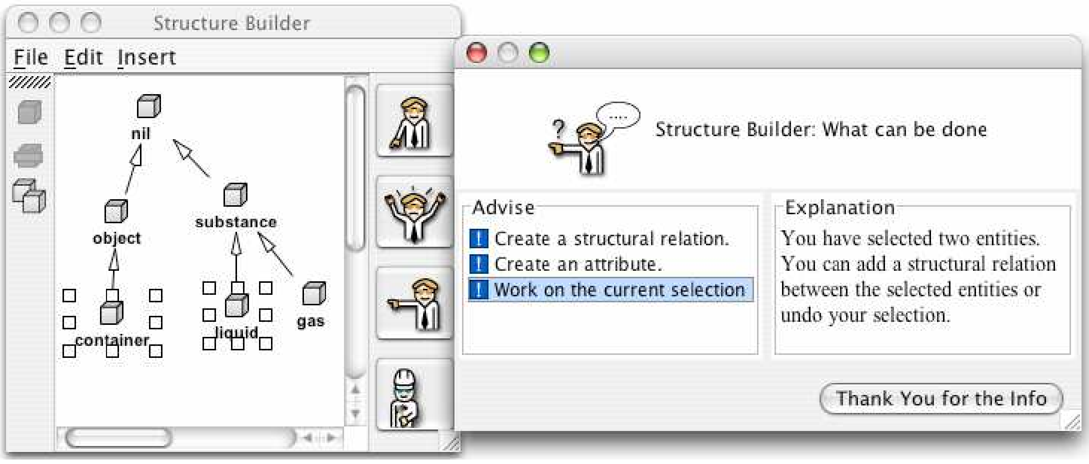


Figure 2: ’What can I do next?’ advice in the context of the Structure Builder

possible subsequent actions. Also context-sensitive help is given which focuses on the specific request for guidance from the learner. For instance, if the learner selects a *quantity* in the model fragment builder and then selects *What can I do next?*, only guidance regarding that primitive will be given. Figure 2 shows an example of help (RHS) given in the Structure Builder context (LHS). In this example the agent gives three advice options (inferred by using a set of rules specifying relationships between model ingredients): ’Work on the current selection’ (because selections are made in the builder), ’Create a structural relation’, and ’Create an attribute’. Notice the latter two are the *only* possible actions a learner can perform in the builder, given what s/he has already created. The learner has selected the first option, and the agent gives an explanation of that (RHS, agent window).

Global feedback on the other hand is based on what the learner has previously constructed in *all* other builders then the one from which the help is requested. The idea is that ingredients are related and must somehow be reused across different builders. If already defined model ingredients are not yet reused adequately, and the reuse might be relevant to the builder from which the help is asked, then the agent will produce an advice on that. Sometimes many advices are possible. We have defined progress levels in order to generate contextual advices associated with each model-building step. Thus, the information gathered enables the help engine to create an ordered list of possible user actions applying to the specific model-building step. Figure 3 shows an example of a global feedback.

4 Design of the experiment

A study was performed with three novices and four experts to assess the usefulness of the multi agent help system and the usability of the MOBUM user interface. The purpose of the novice/expert distinction was not to compare the performance of the two, but rather to ensure that an adequate range of users was covered. For this purpose, the participants were given tasks that corresponded to their capabilities. The task for the novices was to determine the effect of ’food intake’ and ’physical exercise’ on the ’weight of Garfield’. Experts were asked to construct a simulation model of the two-tank system (U-tube). The participants received documentation concerning the assignment, a short explanation of the employed qualitative modelling terms, and a brief introduction to the MOBUM environment. Each session lasted one hour. In both situations (novices and experts) a drawing, illustrating the situation the participants should model, was available in the SWAN SketchPad, the drawing tool of MOBUM.

All computer actions as well as verbal data for each of the sessions were recorded on video. Two types of data were used to evaluate MOBUM: screen information and the verbal

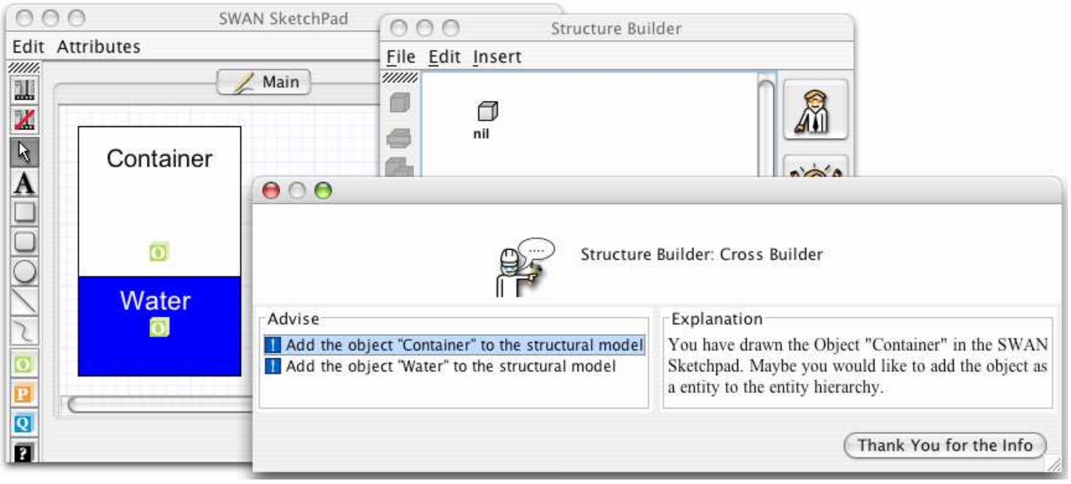


Figure 3: The Cross Builder agent refers to an object in the SWAN SketchPad

utterances of the participants. Participants were asked to think-aloud as much as possible, providing us with valuable information regarding the reasoning underlying the actions taken during the model-building task.

In order to measure the usefulness of the help system, we observed at which moments an agent was requested and if the given feedback was sufficient for clearing the doubts of the subjects about the problem at hand. Additionally, the questions posed by the participants to the experiment leader were analysed to verify whether they were in principle covered by the implemented help system (in which case they could just as well have been solved by the help agents!). While the participant completed each task, the experiment leader noted the number of times an agent was used. In order to measure the participants' performance, the models they created were compared to existing models created by experts.

A second study was performed especially to compare the two model-building environments, MOBUM and HOMER (Table 2). The goal was to evaluate whether the new prototype was more effective and if it would be more appreciated by the users. 28 first-year Psychology students participated in this study. None of the participants had knowledge about building qualitative models as well as about the two systems. The participants were randomly divided into two groups of 14. One of the groups started working with MOBUM for one hour and then changed to HOMER using it for 30 minutes. For the other group the order of the two programs was reversed. The assignment consisted also of building a Garfield model using each one of the two systems. The participants were then asked to fill out questionnaires on MOBUM (QM), HOMER (QH), and a third one on a direct comparison between the two systems (Com).

5 Results and Evaluation

Figure 4 summarises the usage of the agents by the participants. The novices requested help in all the builders and the requested help was of different kinds. Experts on the other hand needed help mainly in the context of model fragments and they accessed the local agent most frequently. This may be explained by the fact that creating a model fragment involves manipulation of all the single model ingredients created previously, as well as determining relations between them. Without exception all novices found the agents useful and essential. The help facility was essential in aiding the participants in solving conceptual problems. For example, a participant wrongly specified *quantities* as *entities* using the Structure Builder. When specifying a model fragment, the participant realised that it was impossible to define *dependencies* between *entities* (they can only be defined between *quantities*). So, the participant backtracked and consulted the agent to understand what had been done wrongly. In doing so, the participant learned what the mistake was.

Another participant had no knowledge about (qualitative) modelling and consequently also no understanding of *points* and *intervals* in a quantity space. But during the process of creating a quantity space, the participant learned about them. It took the participant 15 minutes to specify the first quantity space, 2 minutes for the second, and 30 seconds for the third. In yet another case, after consulting the agents, the participant found the explanation about derivatives and understood their meaning. Later, the participant returned and used the concepts correctly.

The experts did not seem to use the agents to solve problems. When the experts got stuck, they consulted the experiment leader. However, the participants might as well have consulted the agents, as their problems could have been dealt with using the agent-based help facility. Experts seem to use the agents to assess the help potential by trying the help in different situations. However, when trying the agents they got inspired by the advices given. Another support feature frequently consulted was the SWAN SketchPad, the drawing tool of MOBUM, which contained the U-Tube drawing. The participants were consulting the drawing in order to verify if their model included all the details presented in the drawing.

Experts had only a few problems that specifically related

Condition	Questionnaire	Tasks			Questionnaires		
	7 min.	8 min.	60 min.	30 min.	15 min.		
Mobum-Homer	A	reading introduction	Mobum	Homer	QM	QH	Comp
Homer-Mobum	A	reading introduction	Homer	Mobum	QH	QM	Comp

Table 2: Sequence of the questionnaires and tools in the experiment

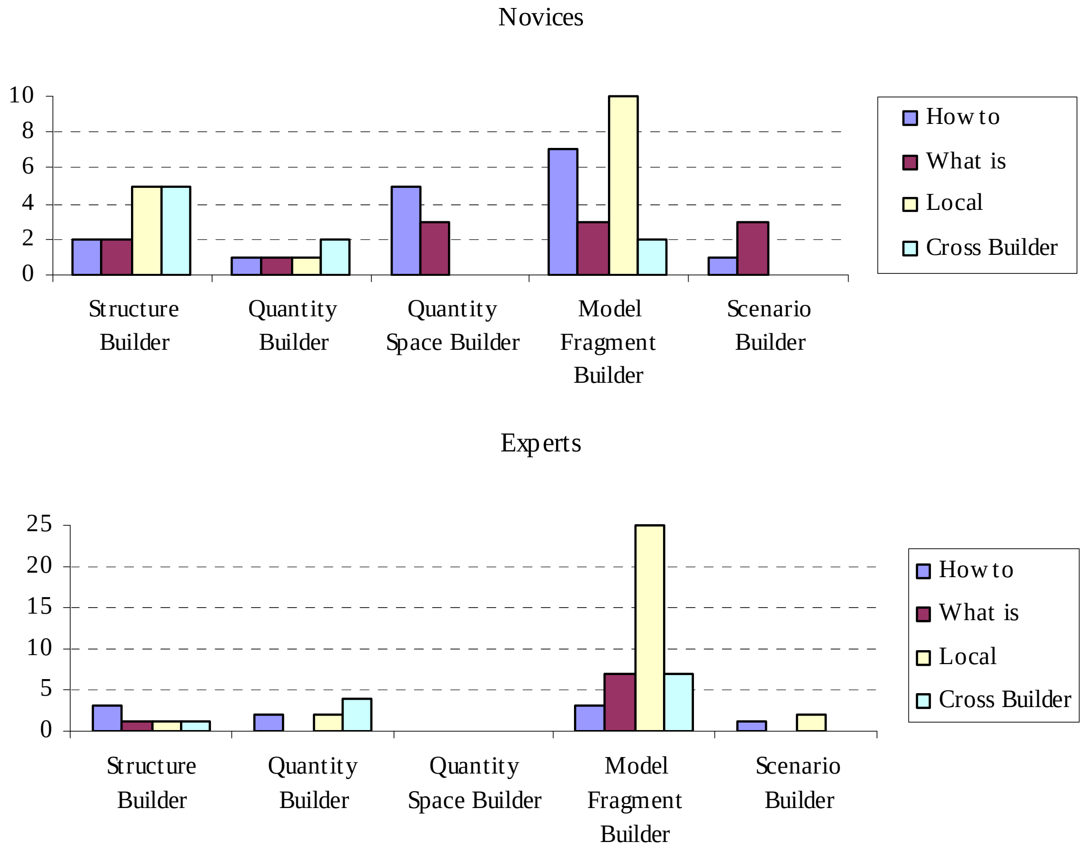


Figure 4: Usage of the agents by novices and experts

to the MOBUM user interface. In our study with HOMER 67 problems were observed while in MOBUM only 10 problems were observed. These results indicate that the features implemented in MOBUM are insightful and effectively support modellers in building their models.

5.1 Results of the Comparison Study

In both situations, HOMER-MOBUM and MOBUM-HOMER there was a strong (and significant) preference for MOBUM over HOMER. For instance, the results of the comparative questionnaire clearly show a significant preference for MOBUM over HOMER ($z=4.4$, $p < 0.0005$). Even when only the first tool ($z=2.7$, $p=0.007$) or the second tool ($z=3.6$, $p < 0.0005$) is measured there was a significant preference for MOBUM over HOMER. A variance analysis was performed to find out if the order had an influence on the results of the experiment. This was not the case. The results for the measure of productivity of both tools did not prove that MOBUM was more effective. We expected that by being more easy to use and giving more guidance, a difference in productivity would emerge. But, there was a high variance among the participants and therefore no strong conclusions can be drawn with respect to this issue. For additional details see [Bessa-Machado, 2004].

6 Conclusions and Discussion

This paper discusses a multi-agent help system that supports learners in building qualitative knowledge models using diagrammatic representations. Being able to create such *conceptual* models (concept maps) may help learners in understanding how and why systems behave as they do. The multi-agent help system is implemented as a part of MOBUM, a workbench for building, simulating, and inspecting qualitative models. The agents are personified and provide context-sensitive help. They provide general support on for instance the model-building ontology, as well as tailored feedback addressing the individual needs of learners.

A study was performed to assess the usefulness of the multi-agent support module. The results are encouraging. Most of the problems the participants encountered were (or could have been) solved by consulting the agents, which reinforces the idea that MOBUM in fact supports the model-building process. A second study was performed to compare MOBUM and HOMER, an earlier developed model-building tool. Due to the large variation in the models created during the experiment we cannot prove that MOBUM is more effective. However, it is safe to conclude that the multi-agent help module effectively influenced the appreciation of the tool: subjects evaluated MOBUM significantly more positive.

Future work could focus on a number is issues. Some initial work has been done on using our model-building workbenches in classroom situations [van der Werf, 2003]. Significantly more effort is needed to actually fit this new approach to science teaching and learner in currently used curricula. Related is the fact that MOBUM is a prototype system. Although it has all the required functionality, it is not fully stable as a software package. For use in classrooms this needs to be addressed.

References

[Bessa-Machado and Bredeweg, 2003] Vania Bessa-Machado and Bert Bredeweg. Building qualitative models with homer: A study in usability and support. In *Proceedings of the 17th International Workshop on Qualitative Reasoning, QR'03*, pages 39–46, Brasilia, Brasil, August 2003.

[Bessa-Machado, 2004] V. Bessa-Machado. *Supporting the Construction of Qualitative Knowledge Models*. PhD thesis, University of Amsterdam, Faculty of Sciences, November 2004.

[Biswas *et al.*, 2001] G. Biswas, D. Schwartz, J. Bransford, and The Teachable Agents Group at Vanderbilt. Technology support for complex problem solving: From sad environments to ai. In K. Forbus and P. Feltovich, editors, *Smart Machines in Education*. AAAI Press/ MIT Press, Menlo Park, California, USA, 2001.

[Bredeweg and Forbus, 2003] B. Bredeweg and K. Forbus. Qualitative modeling in education. *AI Magazine*, 24(4):35–46, 2003.

[Bredeweg, 1992] Bert Bredeweg. *Approaches to Qualitative Reasoning*. PhD thesis, University of Amsterdam, 1992.

[Collins, 1996] A. Collins. Design issues for learning environments. In S. Vosniadou, E.D. Corte, R. Glaser, and H. Mandl, editors, *International perspectives on the design of technology-supported learning environments*, pages 347–362. Lawrence Erlbaum, Mahwah, New Jersey, 1996.

[Forbus *et al.*, 2001] K. D. Forbus, K. Carney, R. Harris, and B. L. Sherin. A qualitative modeling environment for middle-school students: A progress report. In G. Biswas, editor, *Proceedings of The 15th International Workshop on Qualitative Reasoning*, pages 65–72, San Antonio, Texas, 2001.

[Novak and Gowin, 1984] J. D. Novak and D. B. Gowin. *Learning how to learn*. Cambridge University Press, New York, NY, 1984.

[Reif and Larkin, 1991] F. Reif and J. H. Larkin. Cognition in scientific and everyday domains: comparison and learning implications. *Journal of Research in Science Teaching*, (28):733–760, 1991.

[Shimoda *et al.*, 2002] T. A. Shimoda, B. Y. White, and J. R. Frederiksen. Student goal orientation in learning inquiry skills with modifiable software advisors. *Science Education*, 86:244–263, 2002.

[van der Werf, 2003] Rintse van der Werf. The use of qualitative modelling and simulation tools in high school education: an engineering study, 2003.