Investigating the Model Building Process with HOMER

Vania Bessa Machado & Bert Bredeweg

University of Amsterdam

Department of Social Science Informatics

Roetersstraat 15

1018 WB Amsterdam

The Netherlands

E-mail: {vania, bert}@swi.psy.uva.nl

Publication details: Proceedings of the International workshop on Model-based Systems and Qualitative Reasoning for Intelligent Tutoring Systems, pages 1-13, San Sebastian, Spain, June 2nd, 2002, B. Bredeweg (editor).

Investigating The Model Building Process with HOMER

Vania Bessa Machado and Bert Bredeweg
Department of social science Informatics (SWI)
University of Amsterdam
Roetersstraat 15, 1018 WB Amsterdam, The Netherlands
E-mail: vania, bert@swi.psy.uva.nl

Abstract

An experimental study is presented, which investigates the process of building qualitative simulation models using HOMER, a tool for building qualitative models of systems. HOMER consists of a number of dedicated editors aiming at decomposing the complexity of the model building process into more manageable subtasks. The aims of the present study are a) validating the original task analysis underlying the implemented system and the resulting task decomposition, b) identifying problems and/or misconceptions users encounter when building simulation models, c) assessing the tool's usability in supporting the model building process, and (d) defining possible improvements and faults of the present version.

Keywords: Qualitative Reasoning, Building Simulation Models.

1 Introduction

This paper presents the results of an analysis regarding the building of qualitative simulation models using HOMER [1] [2], a modelling tool for building qualitative models of systems, which can be simulated using GARP[3] . In the experimental study reported here, subjects used HOMER for building a model of a U-Tube system [4]. The subject's steps in solving the assignment together with the questions made during the experiment were recorded on video and subsequently transcribed for a fine-grained analysis. The protocols were analyzed and problems, misconceptions and modelling patterns were identified and clustered with regard to the conceptual aspects of the model building activities. Furthermore, problems encountered by participants due to the User Interface were registered for later improvement.

1.1 Building qualitative simulation models

Building qualitative simulation models is a complex process during which a multitude of aspects have to be managed by the model builder. At the most general level the problem of building a simulation model is to specify a set of model ingredients that can be used by a simulation engine to run a simulation in order to produce some required output (usually a particular graph of qualitatively distinct behaviors)[7]. This situation is sketched in figure 1.

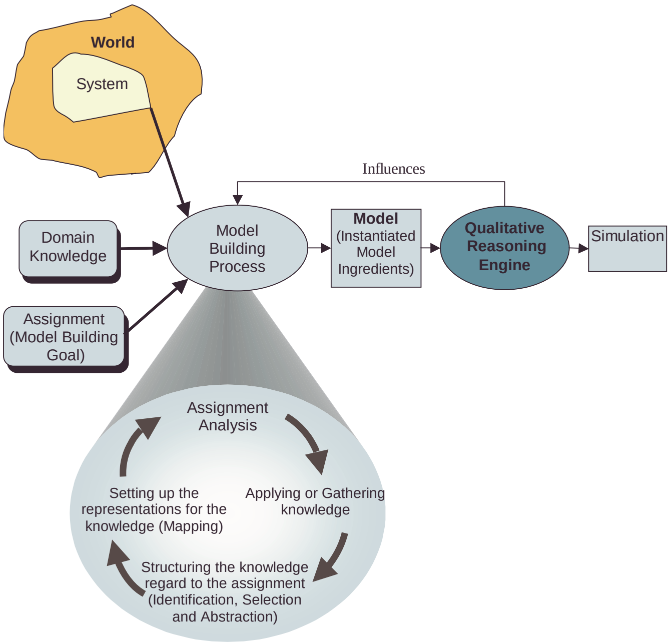

Figure 1: Model Building Process

The figure also shows that a building activity is particularly influenced by two aspects, the assignment and the reasoning engine. The former sets the requirements with respect to *what* must be captured by the model, where as the latter sets constraints with respect to *how* things can be represented in the model. Thus, when we look into the model building process with more detail, we can identify a cycle of activities with two main tasks distinguished. Usually, when initializing the model building process, a modeler analyzes the given assignment (or problem). Next, the modeler applies his (domain) knowledge in order to get the picture of the problem at hand. Certainly, if the modeler doesn't possess (domain) knowledge, s/he must gather knowledge about it before proceeding. Then, the phenomena from the real-world system must be identified, selected and abstracted into a set of system features that must be captured in the model. Finally these features must be mapped into model ingredients that obey the requirements of the simulation engine. Certainly, these activities are not independent.

Following the outline given above, specific aspects can be pointed out as being important for determining the result of a model building activity.

- Experience in model building in general. A model builder who is experienced in identification, selection and abstraction of real-world system phenomena, is in general better equipped to build models (regardless of the particular tool this person has to work with).
- Knowledge of model ingredients. The model building process will be easier, possibly better executed, when the model builder is more knowledgeable of the kind of ingredients that an engine requires.
- Complexity of the assignment (modelling goal). The complexity of an assignment depends on what and how many are the modelling ingredients the users have to work with and how users are supposed to organize/structure these model ingredients. Also, the existing ways of building the model are relevant: building by selecting, building by modifying and building from scratch. See [5] for a

description of these influential aspects.

- Representation of ingredients in a model building tool. Model building tools may differ with respect to how knowledge about system behavior must be represented. The way this is realized will influence the ease of model building.

- Knowledge of tool operation (traditional U.I. issues). The way a tool must be operated may be easy or more complex, for instance because it resembles operations used for other tools (or because it differs significantly from those known tools). Resemblance to known tools will probably make the use of the model building tool easier.

1.2 HOMER

HOMER is a tool for building qualitative simulation models. Technically, the HOMER environment (Figure 2) consists of a set of dedicated editors described as follows:

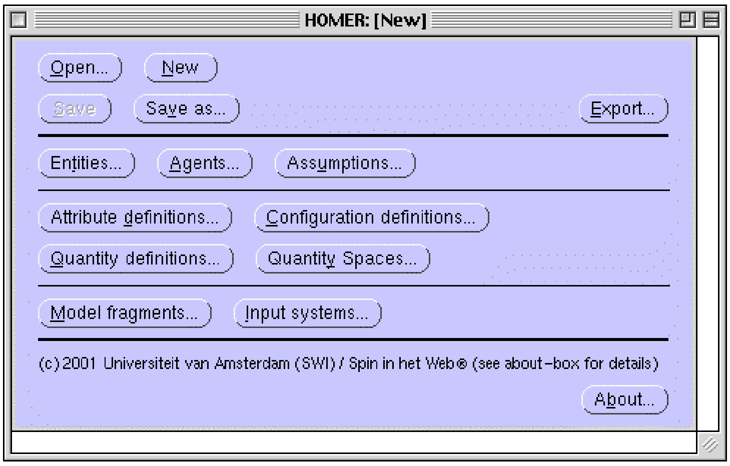

Figure 2: HOMER Main Window

- Entity Hierarchy: In this Editor the user models the (physical) objects that represent the domain. The hierarchical relationships between these objects will be modelled here as well, see Figure 3 for an illustration.

- Attribute definitions: In this Editor the user models the generic attributes that can belong to some physical object.

- Configuration definitions: In this Editor the user models the generic configuration that will be applied between two objects (Structural Relations).

- Quantity definitions: Here the user defines the generic quantities that may be applied to an object.

- Quantity Spaces: In this editor the user creates an ordered set of quantity values that quantities may have. These values are a sequence of alternating points and intervals.

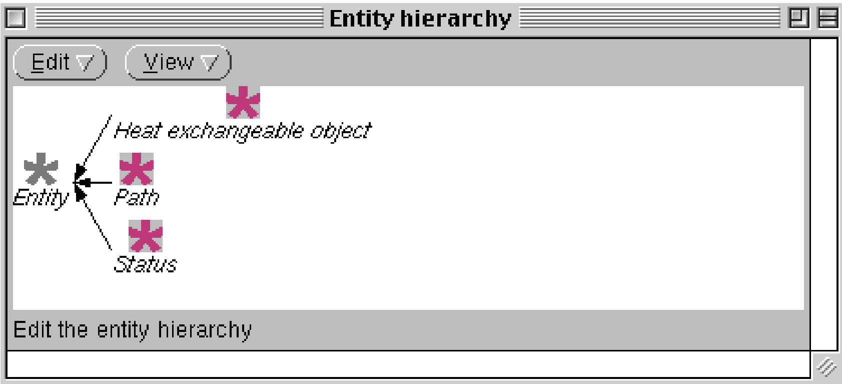

Figure 3: HOMER: Entity Hierarchy Editor

- Model Fragments: In this editor the user constructs the knowledge about the behavior of objects. This includes the specification of features of instances, such as quantities, the values these have, and the dependencies that exist between the quantities. In this editor, the user will be instantiating the generic model parts modelled in the previously mentioned editors (See Figure 4).
- Input systems: In this Editor the user defines the situations that can be simulated. Notice that by definition this can only be a 'selection'(instantiation) of the model parts defined elsewhere in the model. For instance, there is no point in specifying an entity in a scenario that is not used in any model fragment.

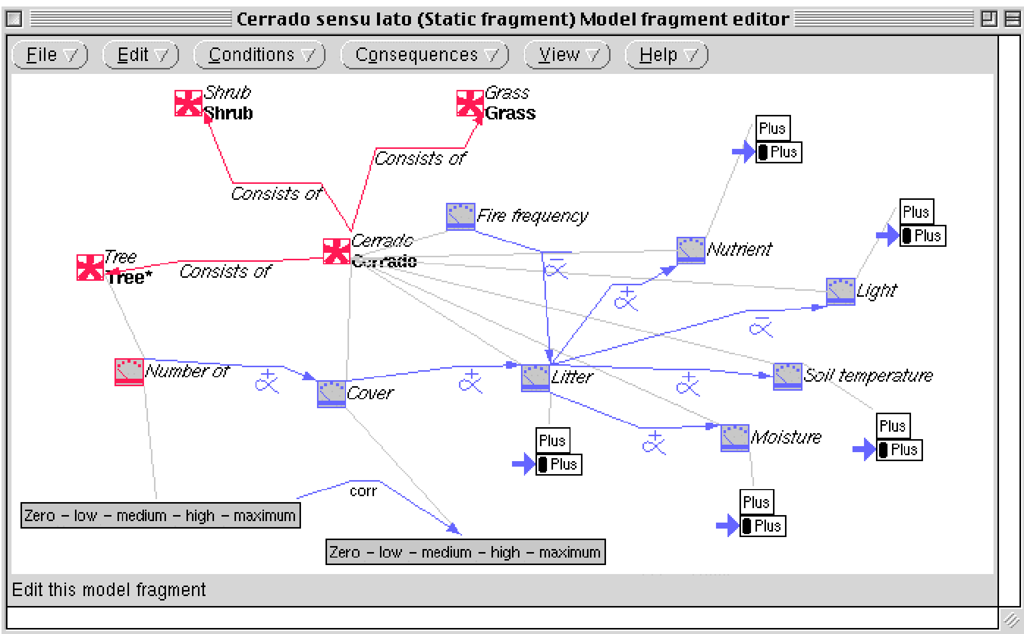

Figure 4: HOMER:Model Fragment Editor

Figure 5 displays the main sequence of usage of the editors. Notice that model fragments and input systems use model parts that come from the editors in the first line, which implies that these model parts have to exist (or have to be created) when creating a model fragment or an input system. It turns out that these model parts are highly interrelated in the context of model fragments and scenarios. For instance, a quantity always belongs to an instance, an attribute always exists between two instances, a proportionality always exists between two quantities, etc.

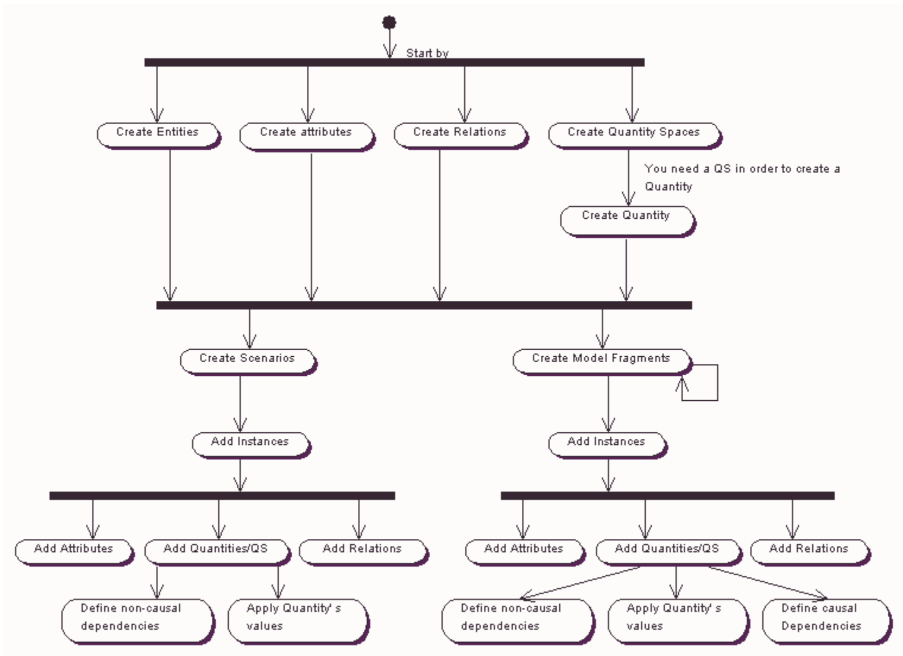

Figure 5: HOMER: Sequence of usage of the Editors

In this experimental study, our aims were a) validation of the original task analysis underlying HOMER and therefore, the resulting task decomposition, (b) discover what problems and misconceptions users encounter when building simulation models, (c) assessing the HOMER's usability in supporting the model building process, and (d) defining possible improvements and faults of the present version.

2 The Experiment

The experiment consisted of two parts. In the first part, the participants subjects should construct a simulation model of a U-tube system from scratch. The experimental setup consisted of handing out to the subjects a documentation containing the assignment, a short explanation of the qualitative modelling terms employed as well as a brief introduction to the HOMER environment. The second part consisted of giving specific tasks to the subjects as well as partial model constructs. All participants should start with the first part of the experiment. After thirty minutes if they didn't have advanced in the assignment of the first part the second part should be introduced.

Each session was recorded on video. The camera was pointed at the computer screen with the purpose of capturing the complete sequence of activities performed by the participants and therewith, gathering information about common behavior of a typical modeler. During the experiment, participants were allowed to ask questions, in fact we encouraged them to do so. Also, during the model building process participants were requested to think aloud. After completing the assignment, the subjects were asked to give a summarizing reflection about the bottlenecks, flaws and possible improvements they encountered while working with the tool. This last step was recorded as well. Each session lasted approximately one hour. During the session, participants could use paper and pencil as well.

Participants

The participants were four people from our department (SWI). Two of them are researchers at our department and the other two are master students. All four participants had had contact with qualitative modelling.

3 Method

This experiment is about evaluating a model-building tool that supports the construction of simulation models. Figure 1 gives an overview of *all* the aspects involved in such a task, while figure 6 focusses just on those directly influenced by the support provided by the tool.

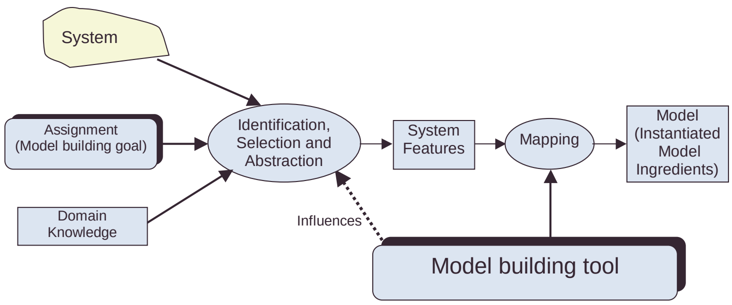

Figure 6: Model Building Tool

The aspects concerning the *Complexity of the assignment* and *Domain knowledge*, which also affect the model building process are not explored in the experiment. The reason for this is that the assignment is the same for all participants and each of the participants already possesses the required *domain knowledge* to solve the given assignment. These aspects therefore have little relevance in the outcome of the experiment. For similar reasons, the *assignment analysis* and *applying or gathering knowledge* steps of the model building process cycle(see figure 1) are also left out.

The doted arrow connecting the model-building tool to the process of identification, selection and abstraction indicates the overall influence that the tool may have on that process. Since the tool is based on a reasoning engine, which uses a specific modelling language, it is natural to expect that during the identification, selection and abstraction of the domain knowledge, the users will somehow be influenced by the possible ways of representing that knowledge.

The arrow connecting the tool to the mapping process represents the strong relation between the two. Following well-known user interface design patterns, the tool provides support by only making available correct modelling primitives, thus constraining the mapping of the system features in such a way that the output always results in a semantically correct model. There are several possible ways in which this support might be implemented. This experiment has thus also been used to evaluate the specific form of support proposed in HOMER. With this purpose in mind, we will firstly look at the task analysis and its resulting products (see Figure 5 for an overview). Secondly, the focus of the evaluation turns to the major point of investigation within this experiment, which is discovering the model building problems encountered by the user while building a system model. Finally, based on important usability principles, we evaluate the User Interface. A discussion and extension of these issues is presented in the sections below.

3.1 Conceptual Aspects of Model Building Activities

3.1.1 Validation of the task analysis

The process of building simulation models involves a series of tasks and subtasks. The end goal of this process is the identification and representation of sufficient knowledge about a system in order to grasp its behavior. The objects found relevant for inclusion into the model and the way they are introduced may differ substantially from modeler to modeler. There are many correct ways of defining a model. Consequently, a multitude of possibly correct task flows may result, differing in the order in which the tasks are realized as well as in the hierarchical refinement into subtask.

In the *Task Analysis Validation* we validate the decomposition of the modelling process into seven major classes[1] (see Figure 5) which reflects the design decisions taken in the implementation of the evaluated tool. Mainly, we wish to know if the tasks and subtasks, and the order in which they occur, match the needs of the typical user.

Secondly, we want to sketch a profile of the typical modeler and understand the way in which the elaboration of a model is usually structured, if the available tasks were correctly interpreted by the subjects and if essential tasks were missing in the proposed task flow.

3.1.2 Model Building Conceptions

Model Building Conceptions focus on the identification of the participants' problems in performing the following (sub)tasks:

1. **Scoping the Model:** Identification of the knowledge relevant for the system at hand (In this experiment the *U-Tube System*). E.g., What are the relevant quantities in the system?

2. **Structuring the Model:** Organization of the model (the relevant knowledge) into a working simulation model which can be used to predict the system's behavior. This step is quite critical. It reflects the participant's view of how to represent the desired system. An example of the questions that might have to be answered at this stage is: How many model fragments are needed to represent the U-Tube System?

3. **Model Building Concepts:** Understanding the concepts that constitute the model of a system. E.g., What is the difference between *attributes* and *quantities*?

3.1.3 Representing The Model

Representing the model refers to representing the system's phenomena in terms of the available ontology for building simulation models. It may happen, for instance, that the modeler knows what he wants to say

[1] Entity, Attribute, Configuration, Quantity, Quantity Space, Model Fragment and Input System

but he doesn't know how to say it using the available ontology. Representing the model (Mapping) plays an important role into the cycle of model building activities. If the user assembled the system features but still doesn't know how to represent the features in the ontology for building simulation models they will not proceed in the cycle. Therefore, a model building tool should help the users in this aspect, such as, by preventing them to perform erroneous mappings.

3.2 User Interface - Overall Usability

In this experiment the subjects were required to go through the process of building simulation models. In order to do so, they had to learn to manage a new tool, making the task more complicated. A proper tool should facilitate the users' comprehension and interaction regarding the solution of the problem. Usability thus, plays a crucial role in attaining one of the major goals in a successful interactive system. The *Heuristic Evaluation* method is used to describe the problems arising during the subjects interaction with the interface. By considering Nielsens ten usability heuristics[6] we interpreted the user's actions in order to infer in how far these actions are related to usability issues in the interface's design.

4 Results

The experiment consisted of two parts. However, it turned out that all the subjects performed satisfactorily on the first part of the experiment, so that the realization of second part became unnecessary. Since the official time to complete the experiment had already past, one participant didn't complete the task of creating an input system (scenario). In the following, the main results of the experiment are summarized and categorized into three main classes: Conceptual Aspects of Model Building Activities (*Task analysis, Model Building Conceptions)*, Representation of the Model and User Interface Evaluation.

4.1 Task Analysis

Sequence of activities

Despite the fact that the existing activities in the highest level of the tool are concurrent (See figure 5), the participants started by creating the hierarchy of entities. Just one participant, Raichu, didn't create a hierarchy of entities. However, he gave meaningful names to the entities when including them in his model fragments. By doing so, we may conclude that he was aware about the existing entities in the U-Tube system but he wasn't aware that the definition of these entities should be done in a hierarchical manner and in a dedicated editor.

In summary, examining the data, we can conclude that, essentially, the mainstream sequence of activities was: Creation of a hierarchy of entities in the first place, followed by the creation of the model fragments and finally the input system. Together, these three concepts constitute the main building blocks in GARP and were also the ones, the subject focused on constantly. Other concepts, such as configurations, quantities and dependencies played a more secondary role as they were more loosely coupled and appeared only in the context of model fragments and scenarios. Although, in the context of Model Fragments and Scenarios, the sequence of performed (sub) tasks was determined mainly by the UI (For instance, the user couldn't add a quantity to the model fragment if an entity, which the quantity belonged to, was not selected.), still there was a mainstream sequence of activities 1)Add Entities, 2) Add Configurations, 3) Add Quantities and 4) Add Dependencies.

A significant deviation of this sequence happened when some *repair* had to be made. Notice that this sequence of activities partially matches the ideal sequence of activities implicitly suggested by the tool. Looking at Fig.5, we can see that, at the highest level we have the (sub)tasks of creating entities, attributes, relations and quantities which matches the sequence in which the participants' completed the various tasks. However it is clear that the creation of Quantity Spaces was not seen as an independent task by any participant as it is suggested by the tool.

Understanding the activity that should be performed

The participant didn't have problems in understanding the activities.

Missing tasks

Although, these were not the main problem in completing the task, some missing tasks were noticed or pointed out.

- Two of the subjects made draws before specifying their models. This can be indicative that a support for this task should be part of the tool
- The subjects pointed out that they missed a complete overview of the model constructed so far. It was mentioned that if they were modelling a complex system it would be harder to understand their models (For instance, because of the crossing lines, hierarchy of Model Fragments, etc)
- An interesting aspect noticed was that the subjects were often constructing a mental model of the behavior of their system, particularly concerning how quantities are causally related. This suggests that the tool should support the causal model view (editor) constructed so far.

4.2 Model Building Conceptions

Scoping the model

None of the participants had difficulties in specifying the U-Tube system's entities. Remarkably, our data analysis shows that all participants experienced difficulties in defining which quantities were relevant for the model. Also, participants effort were in defining the relevant values for a quantity space, defining relevant model fragments and defining when the model building task was finished (all the relevant knowledge was specified). E.g.,

- E.g.:"What quantities to define?
  Maybe we need pressure!"
  "I don't have any quantity. At least we need (quantity) *pressure*. I don't know".
  "Do I need to model *pressure* and *flow*? I need *flow*. Maybe I need *pressure*"
- "I am not sure if I need the value *max*..."
- "Do I need another Model Fragment to define flow?"

- "Am I ready now?"
  "The main problem was in determining when you are finished. You need to be aware of what you have already done and what you still need to do".

Structuring the model

The participants were not sure about *structuring* their models. Sometimes they were unclear about where to add a knowledge item in the model. For instance, a participant was confused about when to specify a quantity's value in a model fragment. Similarly, participants were in doubt about which quantities should be added in a specific model fragment. Still regard to the quantities, participants were in doubt about which quantity space should be given to a quantity and, also defining which entity holds a quantity. E.g.,

- "I didn't define values in the Model Fragment. Do I need to do it for the model fragment be applicable?"
- "I don't know what QS to give to height and I want to say that one height is bigger than another"
- "(Looking at the mf) I don't know if I need the quantity *flow* in this model fragment."
- "I wonder if I need *(quantity) Level* as properties of the container or the liquid entity"

Model Building Concepts

Sometimes the subjects lacked an understanding of the important model building *concepts*. A participant was confused about the meaning of the concepts *attribute* and *quantities*. Another time, a participant wasn't aware of the difference between the conditions and consequences in a model fragment. A participant didn't know the relation between instances in a scenario with the ones in the model fragments. Still, a participant was confused about the difference between two quantities with the same quantity space values. E.g.,

- "Container has *height...*"
  then, the participant selected *Attribute definition*. Therefore, when he should specify the attribute' values the following happened: "Value (?) - I don't know anything about value. I just want to say that the height of one container is bigger than the other." [2]
- Difference between two quantities with the same QS Values
  E.g.:"Are quantities with the same values-name equals?"
- "It was not clear to me that conditions in the model fragment include entities"
- "should the names of entities in the input system match with the names given in the MF?"

4.3 Representing the Model

The results also show that the subjects had problems in specifying the knowledge using the available ontology. During the experiment, a subject tried to add a configuration (relation) between two quantities. This shows that there was a confusion of the meaning of configuration once that this can be done just

[2] At this point the experiment supervisor intervened and explained that what he meant was a quantity

between entities. Also, in some cases, subjects did not understand the conceptual difference between an inequality and a proportionality. Some examples are given below.

- "How do I say that there is a Flow due to the difference in the amount of water?"
- "I want to represent that *PressureDifference* is the difference of the two (quantities) *levels*."
- "Now I expect proportionality. I mean the pressure difference is proportional to the level difference"

4.4 User Interface

This section summarizes the evaluation by emphasizing the more significant usability problems. The evaluation and judgement of each concern or problem is done in compliance with Nielsens ten usability heuristics.

4.4.1 Visibility of the system's status

The "New" button in the main window mislead two of the subjects. As the system doesn't show that the user is working on a new model when it is initialized, the subjects wondered what the status of the system was.

Inside the Quantity Builder, when intending to create a new quantity, while all subjects typed in the quantity's name into the name field, they failed to realize that the quantity would only be created by pressing on the "Add" button. It sometimes happened, that the user thought s/he was creating another quantity, when s/he actually was editing the previous one.

4.4.2 Match between system and the real world

In the main window the menu selections matched the terms commonly used by the subjects. The "Configuration definitions" option was one exception. This option enables users to specify relations between entities but it was not clear to the subjects.

Usually, when specifying a correspondence, it should suffice if one says that Quantity A corresponds to Quantity B. In HOMER, matters are somewhat complicated by the fact that a correspondence has to be specified by the selection of the associated Quantities' Quantity Spaces. This feature also was confused by the participants.

4.4.3 User Control and Freedom

The user must use the system as it is, there is no customization available.

4.4.4 Error Prevention

The *Error Message* in the creation of a Quantity without an associated QS confused the subjects. Especially after having invoked the QS Editor from within Quantity Editor in order to create the quantity's quantity space. They reckoned it should have been automatically associated to the Quantity that was being created. Frequently, this misunderstanding lead them not to finish the task or to loose the quantity entirely.

When adding an object to a model fragment or scenario the last added element is always selected by default, which makes the subjects to add some elements to the wrong font.

4.4.5 Consistency and Standards

Looking at the Editors from the highest level, only in the Quantity Editor the user cannot create an object *per se*. In order to create a quantity it must be associated with a Quantity Space. This caused a confusion among several of the subjects.

In the Entity Editor one of the subjects expected a ”New” button in order to create a new Entity. There is none. Instead there is the ”Add Child” button. At the same level in the others editors, however, the subjects noticed that there is a ”New” button. According to them this was not consistent.

4.4.6 Recognition rather than recall

When adding objects (attributes, configurations and quantities) to the Model Fragments or to the Input System, the user has the option of editing these objects inside their original editors. However, with the resulting proliferation of windows, the users would get lost and not know exactly at which step of the creation of their model they were. Also, the distinction between the Editor and the Creator was not always apparent.

4.4.7 Flexibility and efficiency of use

None of the users realized how to proceed in order to set up a value for a quantity.

4.4.8 Aesthetic and minimalist design

In a Model Fragment and Input System, the option of hiding and showing information wasn't explored by the subjects. Therefore, when adding a quantity all information concerning it (Quantity Space and Derivative) was also added to the window even if it was not relevant. Moreover, two of the subjects complained about too much information on the screen.

4.4.9 Help users recognize, diagnose, and recover from errors

Most of the errors were solved by repeated trial and error attempts. The system is rather deficient in providing help at these times. On several occasions the experiment's supervisor had to intervene in order for the subjects to continue. A subject selected three objects to add a dependency to. There isn't any help in regard to this error.

4.4.10 Help and documentation

A printed version containing Help about the system's functionality and model building terms was available to the subjects. However, none of them did consult it.

5 Discussion And Future Directions

Our major goal in performing the experiment presented in this paper was in investigating the main problems users encountered when building simulation models. Additionally, we wanted to validate the task analysis underlying the implemented system and the resulting task decomposition.

HOMER provides all necessary modelling primitives as well as support to assist the user in the model building process. Before HOMER, the only way to build simulation models (in the context of GARP simulation engine) was by using text files. In this way, a lot of expertise is demanded from the builder not only in formalizing his model correctly but also in taking care of syntactic details such as comas, parenthesis etc. HOMER allows graphical representation of concepts and their relationships. In addition, it provides the means of representing knowledge in a simple and intuitive visual form. The results shows that the visualization tools help users focus on performing a specific task.

Based on the patterns in participants data noted above, two features will be the focus of our future research. First, the design of a support module that can guide the users through of the model building process. The support module is intended to reduce the cognitive load as well as to give to the user more confidence in the process of model construction. To be used in educational settings, the support module should incorporates support in terms of model content and also knowledge about the system status. It means, being able to keep track of the users actions, reasoning about them and providing feedback to the users. Second, the design of a flexible model building tool that can support and maintain intermediate models for the user and, also providing the user with a global overview of certain model parts. For instance, showing how all the model fragments that have been created will interact and also, it was apparent in the results that is helpful to provide the user with a 'causal model viewer'. This would allow the user to investigate if and how the causal dependencies, that have been defined in the different model fragments, are related (thus, without running the simulator first). Additionally, all the improvements related to the user interface usability and functionality should get attention in a new design.

References

[1] Homer 2.0 gebruikers documentatie. Technical report, SWI/UvA, 2001.

[2] Homer 2.0 technische documentatie. Technical report, SWI/UvA, 2001.

[3] B. Bredeweg. *Expertise in qualitative prediction of behaviour.* PhD thesis, University of Amsterdam, Amsterdam, The Netherlands, March 1992.

[4] K. D. Forbus. Qualitative process theory. *Artificial Intelligence*, 24:85–168, 1984.

[5] V. Bessa Machado. Towards understanding model building complexity. Technical report, SWI/UvA, 2001.

[6] J. Nielsen. *Usability Engineering.* Academic Press, San Diego, 1998.

[7] C. Schut and B. Bredeweg. Supporting qualitative model specification. In *Proc. of 2nd International Conference on Intelligent Systems Engineering*, Technical University of Hamburg, Hamburg, Germany, September 1994.