



Project no. **004074**

Project acronym: **NATURNET-REDIME**

Project title: **New Education and Decision Support Model for Active Behaviour in Sustainable Development Based on Innovative Web Services and Qualitative Reasoning**

Instrument: **SPECIFIC TARGETED RESEARCH PROJECT**

Thematic Priority: **SUSTDEV-2004-3.VIII.2.e**

D4.1

Single-user QR model building and simulation workbench

Due date of deliverable: 25/11/2005 (Phase 3)
Actual submission date:
Software phase 1 and 2 available since: 01/09/2005
Software phase 3 available since: 01/03/2006
Written summary documentation (phase 3): 08/04/2006

Start date of project: **1st March 2005**

Duration: **30 months**

Organisation name of lead contractor for this deliverable:
University of Amsterdam¹

Revision: Final

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

¹ AUTHORS: Bert Bredeweg, Anders Bouwer, and Jochem Liem

Abstract

The goal of task 4.1 is to construct a seamless integration of the Qualitative Reasoning software (increasing usability) that will be used in the NaturNet-Redime project. This software consists of GARP for simulating, HOMER for building models that can be simulated by GARP, and VISIGARP, which provides graphical means to control and inspect simulations generated by GARP. This document presents the software that has been created for this purpose.

Document history

Version	Status	Date	Author
01	Draft following phase 2	15/09/2005	Bredeweg
02	Draft of 4.5, 4.6, 4.7, pre-draft of 4.1	04/11/2005	Bouwer
03	Final version describing the phase 2 version of the Garp3 workbench.	07/11/2005	Bredeweg
0.4	Small corrections	08/11/2005	Bouwer, Bredeweg
Final	Final version, corrections and including the additional workbench details implemented in phase 3.	08/04/2006	Bredeweg

Contents

1	INTRODUCTION	5
2	REQUIREMENTS.....	6
3	PLANNING AND STATUS.....	8
4	INTEGRATION OF EXISTING SOFTWARE	9
4.1	PREVIOUS SOFTWARE.....	9
4.2	INTEGRATED SOFTWARE: GARP3.....	10
5	ACTION ICONS AND TOOL-TIPS	12
6	MODEL INGREDIENT ICONS AND TOOL-TIPS	17
7	TOOLBARS	19
8	EQUATION AND VALUE HISTORY	23
9	SIMULATE AND BUILD INTEROPERABILITY	25
10	SAVED SIMULATIONS	26
11	ADDITIONAL FEATURES	27
12	TRACE OPTIONS AND SIMULATION PREFERENCES.....	29
13	CONCLUSIONS	31
14	BIBLIOGRAPHY	32

1 Introduction

The objective of Work Package (WP) 4 is to develop qualitative reasoning software that is easy to use for building and examining qualitative models, and thus facilitates learning. To achieve this, WP is divided into 4 tasks:

- **Task 4.1 Single-user Qualitative Reasoning Workbench**
Enable knowledge capture using state-of-the-art Artificial Intelligence technology (qualitative modelling and simulation) by people who have no detailed knowledge of that technology.
- **Task 4.2 Frequently Asked Questions (FAQ) and Debug Support**
Disseminate experience and knowledge available within the Artificial Intelligence community concerning qualitative modelling and simulation in a format that fits the needs and level of understanding of the partners in the project and the external users.
- **Task 4.3 Collaborative Qualitative Reasoning Workbench**
Facilitate collaborative modelling and simulation using Web-based software to effectively store, share, and re-use qualitative models and model parts among participants.
- **Task 4.4 Intelligent help and support system**
Intelligent help, to coach users in knowledge discovery using the QR software developed in the project.

This document presents the results of task 4.1, the 'Single-user Qualitative Reasoning Workbench'. The organisation of this document is as follows. Section 2 enumerates the main requirements for the to be developed software. Section 3 summarises the planning and status. The important features that have been realised in the new software (Garp3) are discussed in the remaining Sections.

2 Requirements

The goal of task 4.1 is to reshape the QR software (increasing usability) that will be used in the NaturNet-Redime (NNR) project and by doing so enable knowledge capture using state-of-the-art Artificial Intelligence technology (qualitative modelling and simulation) by people who have no detailed knowledge of that technology.

Three tools (software packages) will be used in the NNR project:

- GARP for simulating (Bredeweg, 1992; Linnebank, 2004).
- HOMER for building models that can be simulated by GARP (Jellema, 2000).
- VISIGARP, which provides graphical means to control and inspect simulations generated by GARP (Bouwer, 2005).

Although these three tools are all implemented using the programming language SWI-Prolog², they are three different programs. To simplify the use of qualitative technology within the NNR context, these tools need to be integrated into a seamless environment that provides all the functionality as if it were a single program. Sub-activities include:

- Integrated user interface
- Increasing usability
- Scenario manipulation in simulation mode (setting initial quantity values)
- Import and export of files storing models
- Simple copy/paste (within a model)
- Repair of currently known bugs and problems
- Providing a set of well known model examples
- Bug-report and help-request facilities

Following these initial requirements, the work in task 4.1 was divided into three phases. Phase 1 concerns the basic integration of the existing software packages. Phase 2 concerns improvements to the interface of the integrated software. Phase 3 concerns additional means to provide users with control and feedback on their modelling and simulation activities. The three phases include the following aspects:

- Phase 1: Integration of existing software
 - Technical integration (make one software package)
 - Assign assumptions to entities
 - Fix current known bugs
 - New dependencies (e.g. dq_correspondence)
- Phase 2: User interface improvements
 - Graphics for action icons
 - Graphics for model ingredients
 - Build button bar
 - Scenario and model fragment editors, and dependency screen
 - Tool-tip
 - Main screen, scenario and model fragment editors, and dependency screen
 - Simple copy/paste for model fragment and scenarios
 - Debug support
 - Save simulations

² <http://www.swi-prolog.org/>

- Phase 3: User support and control options
 - Help button
 - Conditional model fragment refinement
 - Default functions for generating quantity behaviour (e.g. generate all values for Q1)
 - Model fragments overview as subtype hierarchy, conditional for hierarchy, and on/off switch
 - Simulation preferences
 - Engine trace formatted and made available
 - Debug support – meta data
 - Options to document model and problem features
 - Export screen diagrams as EPS documents for scenario and model fragment editors, and dependency screen
 - Technical documentation

The requirement specification is based on research, carried out by the UvA, on the use of QR software by potential end-users (e.g. Bessa Machado, 2004; Groen, 2003; Haverkort, 2000; Tjaris, 2003; and Werf, 2003). The full requirements specification document (M4.1 – Requirements analysis and re-design for single-user workbench) is in Dutch³ and available on request. The new software is called: Garp3.

³ This is due to the fact that the contract and regulations used by the company 'Spin in het Web' are in Dutch. As a consequence the 'requirements specification' and the 'contract' between 'Spin in het Web' and the University of Amsterdam are also in Dutch.

3 Planning and status

The requirement specification, design, and testing of the workbench is carried out by the University of Amsterdam. The implementation and debugging is done via subcontracting⁴ the Dutch company 'Spin in het Web'⁵ who is highly experienced in using SWI-Prolog and XPCE, the software in which the current QR tools are implemented. This company also has expert knowledge on qualitative reasoning. The contact between the University of Amsterdam and Spin in het Web is in Dutch.

Due to the late notification of contract signing (only a few days before the new month started) it became impossible for the subcontractor to start on March 1st, 2005. The subcontractor was also not fully available during June 2005 and July 2005. This resulted in a delay of the final product delivery (D4.1). However, in order not to cause any delay for the rest of the project, a fully functional workbench was scheduled (and delivered) for September 9th, 2005 (result of phase 1 and 2). The final version of the workbench was planned for November 25th, 2005 (result of phase 3), and delivered March 1st 2006. Although strictly speaking the phase 3 software is delivered late, compared to the initial planning, the current situation is actually beneficial for the NaturNet-Redime project. The advantage being that partners have effectively more time to test the software and report bugs (if any). Moreover, the warranty period provided by the subcontractor overlaps with the actual use of the software by the work package 6 partners. This means that if technical problems occur that the subcontractor is still available to repair them.

Work package 6 partners working on the case studies start using the software fully after they have delivered D6.{2,3,4}.1 (which was due February 2006). To prepare these partners for this, and also to coach them in creating D6.{2,3,4}.1, a workshop was organised in Amsterdam (October 11-14th, 2005). During the workshop users worked with the phase 2 version of the Garp3 software. No significant technical problems appeared during this meeting and all the required functionality turned out to be available. A second workshop has been organised in Sofia (March 27-31st, 2006). One of the goals of this workshop was to further deepen the knowledge and expertise of the work package 6 partners in using the Garp3 workbench. During this workshop the phase 3 version of the software was used, and again no significant malfunctioning occurred. This is encouraging for the expected functioning and stability of the final release of the Garp3 workbench.

⁴ For details see NaturNet-Redime technical annex Section A.2.

⁵ <http://spininhetweb.nl/>

4 Integration of existing software

4.1 Previous software

The status quo consisted of three tools implemented in part by different people to address different needs:

- Garp: producing qualitative predictions for scenarios when given a library of model fragments (Bredeweg,1992; Linnebank, 2004). The main screen of this package is shown in Figure 1.
- VisiGarp: visualisation and control of simulation results (Bouwer, 2005). The main screen of this package is shown in Figure 2.
- Homer: supporting model building (Jellema, 2000). The main screen of this package is shown in Figure 3.

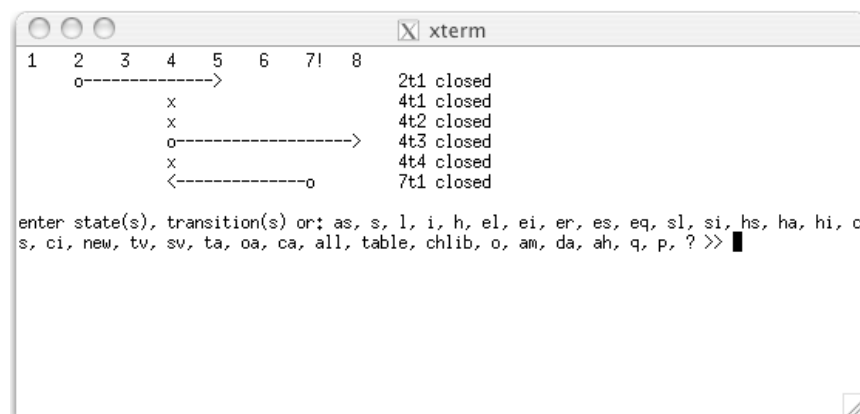


Figure 1: Garp2 command based interface

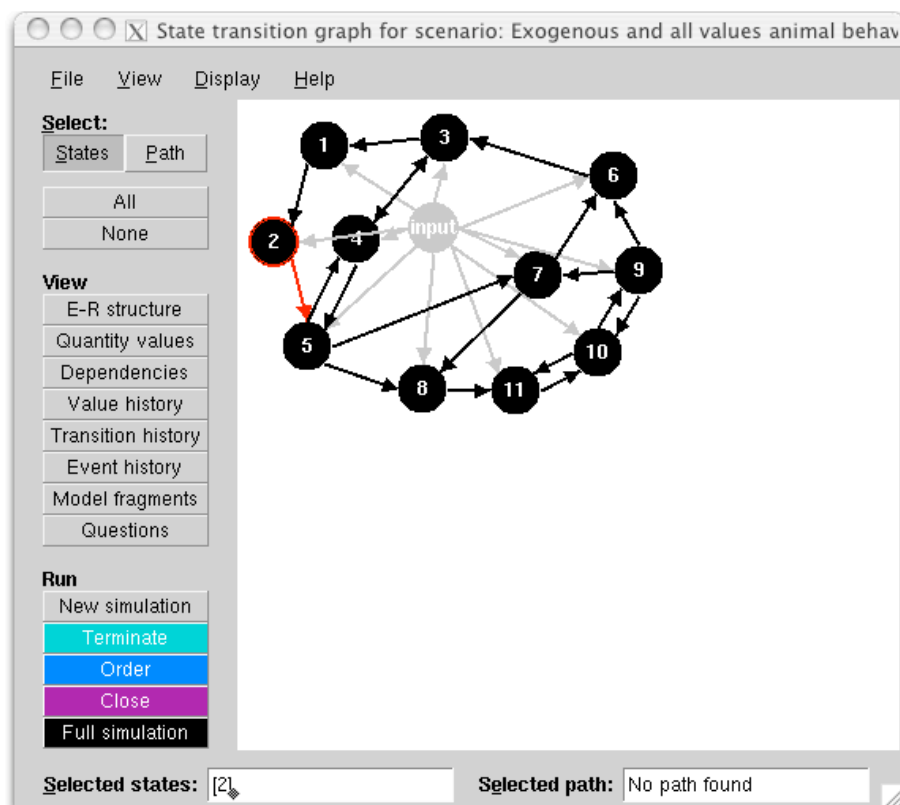


Figure 2: VisiGarp2 user interface for running and inspecting simulations with Garp2

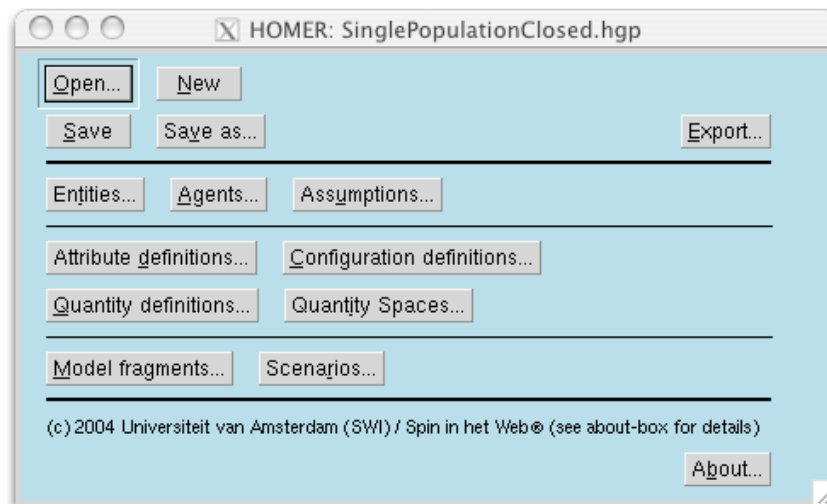


Figure 3: Homer2 user interface

While all three software packages are based on the Garp framework for Qualitative Reasoning and Modelling (QRM) and therefore make use of essentially the same basic representation format (a Garp model consists of a collection of logical predicates, implemented in SWI-Prolog), each of these tools uses different (graphical) representations in the interface. Moreover, in order to use the full power of both modelling and simulation a user has to switch between these applications continuously. Switching not only means to put a different piece of software ‘in front’, but also to explicitly export and import model files between the applications every time a model ingredient is changed. This is of course not a very efficient way of using the QRM potential, and hampers the applicability and uptake of the available tools.

4.2 Integrated software: Garp3

Integration of these three tools should lead to one new tool that incorporates all of the original functionalities, and thus incorporates the advantages of each tool, but adds interoperability and an easy to use uniform user interface. Issues to be addressed in the integration:

- Organizing all functionalities in an integrated workbench, and provide access to the key functions from the main screen.
- Uniform use of graphical icons, diagrams, and colour
- Uniform use of vocabulary
- Making efficient use of having different functionalities in one application (interoperability)

Figure 4 shows the main screen of the new Garp3 workbench after the technical integration (result of work carried out in task 4.1 phase 1). Figure 5 shows the main screen of the new Garp3 workbench after creating and using a uniform vocabulary and graphical user interface (result of work carried out in task 4.1 phase 2). Figure 15 shows the main screen of Garp3 workbench following phase 3.

Also part of the integration task was to fix known bugs in either of the software packages including:

- Make top node entity hierarchy editable.

- The change requestor should consider *all* structural relations between entities, agents, and assumptions before classifying model fragments as 'incomplete' due to missing structural relations.
- Prohibit the use of numbers as names for quantities.
- Sort model fragment in lists (in the simulate context) in alphabetic order.
- Distinguish properly between multiple occurrences of the same model fragment type as conditions, e.g. allow quantities with the same name but belonging to different conditional model fragments to have different values.
- Fix incorrectly cleaning and removing of configurations in the original model fragments when more than one of these model fragments are used together as conditions in a some other model fragment.
- Allow entities to also have 'private' assumptions that only apply to a specific entity (in addition to having assumptions that hold for all entities).
- Optimise selection of 'all states' in simulate so that such selections also happen within reasonable time for larger simulations (e.g. in the case of hundred states or more).
- Uniform (and/or add) window scaling functionalities throughout the software.
- Place quantity space (magnitude and derivatives) icons always close together when a quantity is added to an entity.

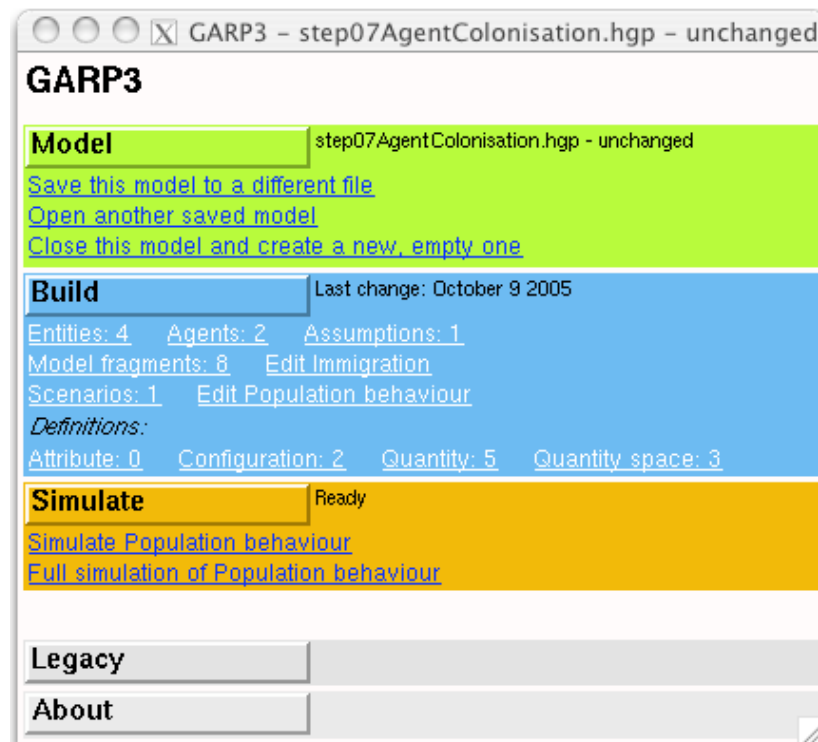


Figure 4: Garp3 user interface after technical integration of the software (following phase 1)

5 Action icons and tool-tips

To increase the usability of the workbench a graphical language has been developed from which icons have been derived to signify the meaning of the underlying vocabulary⁶.

As the vocabulary used in qualitative reasoning and modelling is in principle unknown to non-computer scientists it is impossible to develop a set of action icons that is immediately understood by such users. By definition, the vocabulary and the corresponding icons have to be learned by users. Therefore, more important than immediately 'understanding' the meaning of an icon, is the ability for users to learn the meaning and remember the meaning for future use of the software. To support this process the action icons are organised in a structure that highlights their meaning as much as possible. Table 1 shows one of the key icons of the vocabulary, the 'blue earth circle'. This icon refers to the *model* that is processed using the workbench at a certain moment. There are three types of manipulations that a user can perform: *filling* a model (save, open, etc.), *building* ingredients to compose a model (create entities, quantities, etc), and *simulating* a model (running a specific scenario, inspecting the generated causal model, etc).

Table 1: Model, model ingredients, and model simulation

		
File: Model	Build: Ingredient	Simulate: Model

Following this basic decomposition, the main screen of the workbench is divided into four areas: one for each of these groups of manipulations and one for additional options such as 'help' and 'about'. For each group of manipulations additional icons have been developed. Table 2 shows the manipulations that are accessible from the main screen. A screenshot of the main screen of Garp3 is shown in Figure 5⁷.

Table 2: Garp3 interface – Main screen action buttons

					Help
File icon	Model status info				Garp3 logo
F1	F2	F3	F4	F5	F6
Build icon	Build status info				
B1	B2	B3	B4	B5	B6
B7	B8	B9	B10	B11	B12
Simulate icon	Simulate status info				
S1	S2	S3	S4	S5	S6
					About

There are six file manipulation actions. Two of those concern legacy mode. Legacy refers to Garp2 models, previously created without graphical aid. Garp3 is thus upwards compatible and allows for old models to be imported and simulated. The *file* actions and accompanying tool-tip text are:

⁶ This document discusses the main ideas behind the user interface and the Garp3 functionality. For a full description see D4.2.1 (User-manual for single-user version of QR workbench).

⁷ To benefit from graphics and text, the tool-tip text is always shown when the mouse is on top of an icon (see also Figure 15).

- F1: Open model from file
- F2: Save current model to file
- F3: Save current model to new file
- F4: Start new model
- F5: Open model in legacy mode
- F6: Save model in legacy mode



Figure 5: Garp3 user interface using uniform vocabulary and graphics (following phase 2)

There are twelve action buttons related to creating model ingredients. Two of those, B8 and B10 are short-cuts, which allow a user to directly further edit the last edited scenario or model fragment, instead of having to open the list of scenarios, or model fragments, and select one to work on. This is a very useful feature that significantly reduces the number of mouse clicks (1 instead of 3) for two of the most used activities. B6 refers to a dedicated editor that can be used to add comments and meta-data about the model and the model builder. The *build* actions and accompanying tool-tip text are:

- B1: Open entity hierarchy editor
- B2: Open attribute definitions editor
- B3: Open configuration definitions editor
- B4: Open quantity definitions editor
- B5: Open quantity space definitions editor
- B6: About this model (Meta-data about model and maker)
- B7: Open scenarios editor
- B8: Edit last changed scenario
- B9: Open model fragments editor
- B10: Edit last changed model fragment

- B11: Open agents hierarchy editor
- B12: Open assumptions hierarchy editor

There are six action icons related to simulating a model. S2 and S3 provide short cuts to the scenario simulated last. This is again a very useful feature that significantly reduces the number of mouse clicks for these often used activities. S5 opens the trace window in which a user can select options to follow the inferences made by the engine. With S6 the user can specify preferences concerning the way the engine reasons (e.g. to set the 'closed-world assumption' to 'on' or 'off'). The *simulate* actions and accompanying tool-tip text are:







- S1: Select a scenario to simulate
- S2: Simulate current scenario
- S3: Full simulation
- S4: Open the simulator to its current state or a saved simulation
- S5: Open trace window
- S6: Simulation preferences

Finally, there are three general buttons, namely:

- Help: opens a browser with help on QRM (see also Task 4.4)
- Garp3 logo: opens a browser with general information about Garp3
- About: provides information about the software developers

Following the initial icons for 'model' and 'model ingredient' further icons are used to signify the different ways in which models and ingredients can be manipulated. Some of the typical icons are shown in Table 3. Figure 6 and 7 illustrate how these icons are used in the context of a specific model building activity. In this case the selector for model fragments (see below for further explanation).

Table 3: Model and model ingredient action buttons

					
Save ingredient in model	Delete ingredient from model	Create new ingredient	Show ingredient properties	Copy ingredient	Delete ingredient

Reading from top to bottom the action buttons in the model fragment selector can be used to do the following:

- Add model fragment
- Edit selected model fragment
- Show model fragment properties
- Copy selected model fragment
- Delete selected model fragment

Also available are actions buttons for (further explained later in this document):

- Make selected model fragment inactive/active
- Show default view
- Save current view
- Open another view
- Show parent-child relations (on in Figure 6, LHS)
- Show conditional relations (off in Figure 6, LHS)

Notice that, the option of copying model fragments, and scenarios is new in Garp3. These options greatly reduce the amount of work that modellers need to do in order to

create model fragments and scenarios. Although for the user this option appears to be 'only' an action button in the interface, the underlying reasoning is rather complex. When copying a model fragment for instance, the engine should correctly take into account all the model ingredients defined in the model fragment that is copied and assure proper occurrence of those in the context of the new model fragment.

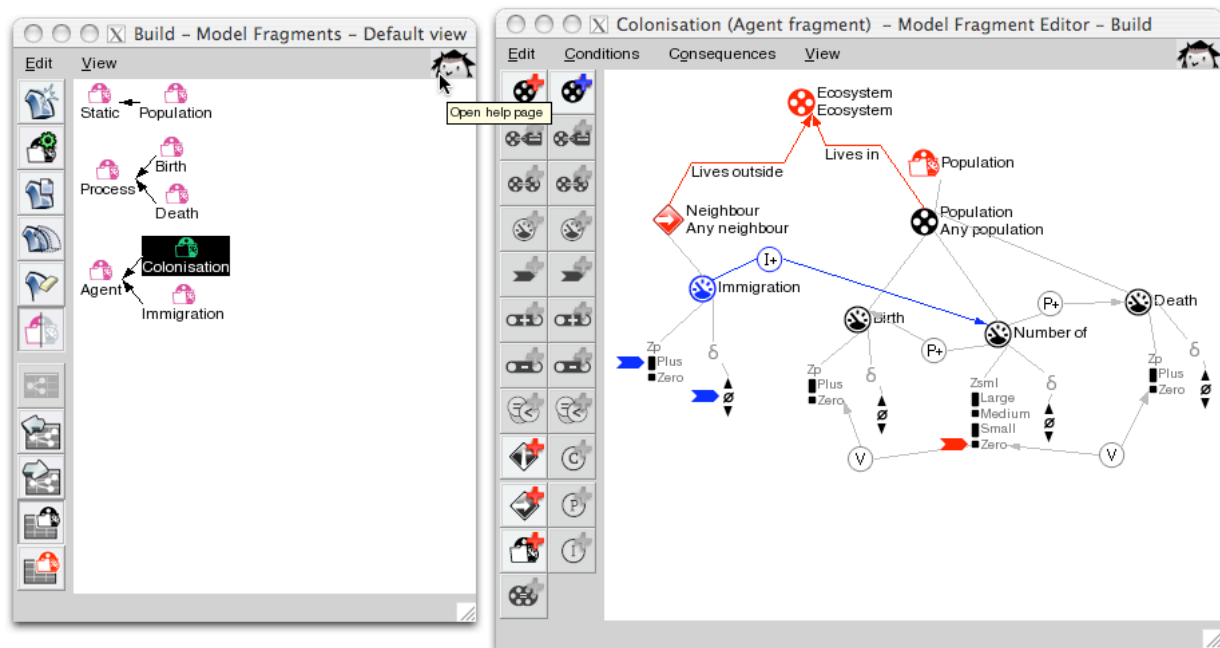


Figure 6: Action buttons in the model fragment selector (LHS). From this selector fragments can be opened for editing as is done for Colonisation (RHS). The figure also shows the help-button tool-tip text for the model fragment selector.

Figure 7 shows the editor for creating quantity spaces and depicts a similar use of the icons discussed above, albeit in a different context.

Reading from top to bottom (and from left to right) these icons can be used to do the following:

- Add quantity space
- Copy selected quantity space
- Delete selected quantity space
- Add value high
- Delete value high
- Split interval
- Combine intervals
- Add value low
- Delete value low
- Save changes
- Delete changes
- Close window

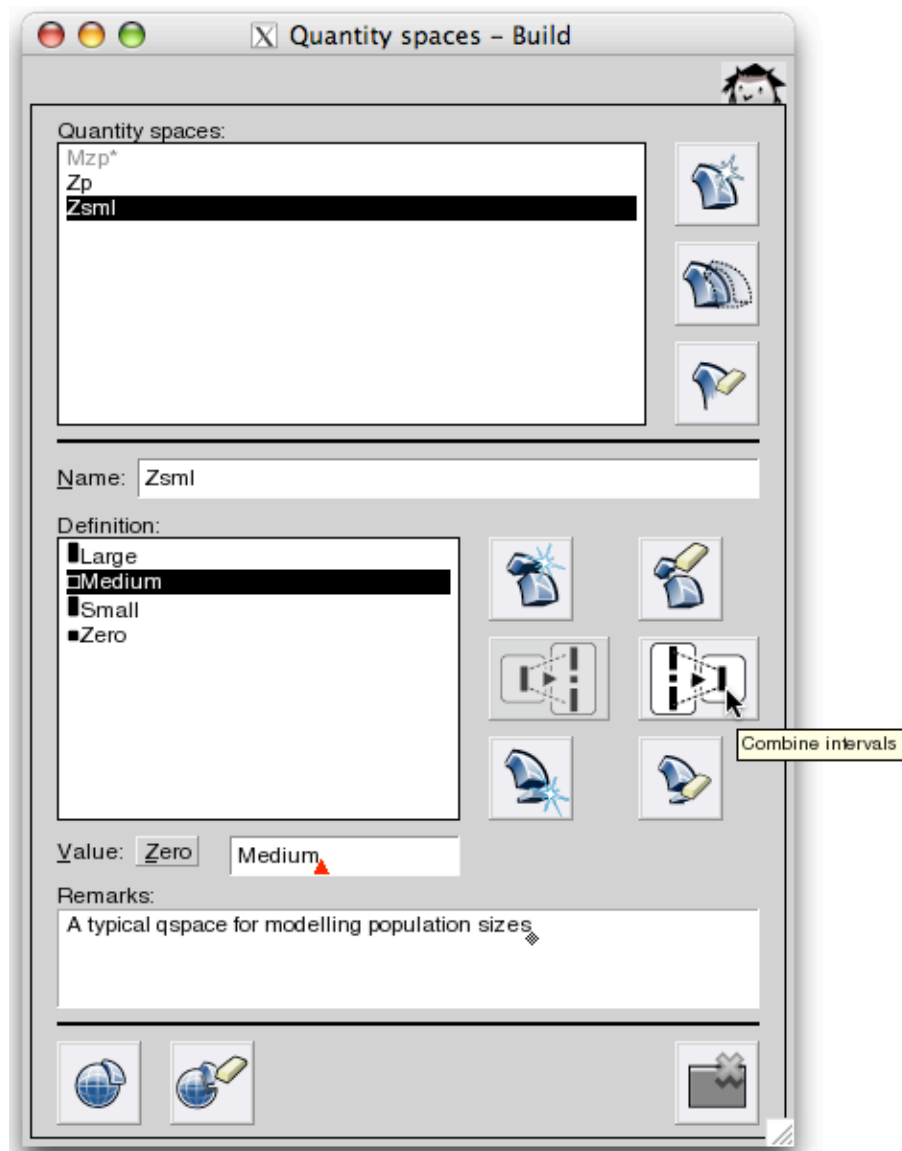


Figure 7: Action icons in the quantity space editor

6 Model ingredient icons and tool-tips

The graphical icons used to signify the model ingredients have been redesigned in the new software. Figure 8 illustrates some of the new icons. Table 4 enumerates the meanings for some of the icons used in this figure.

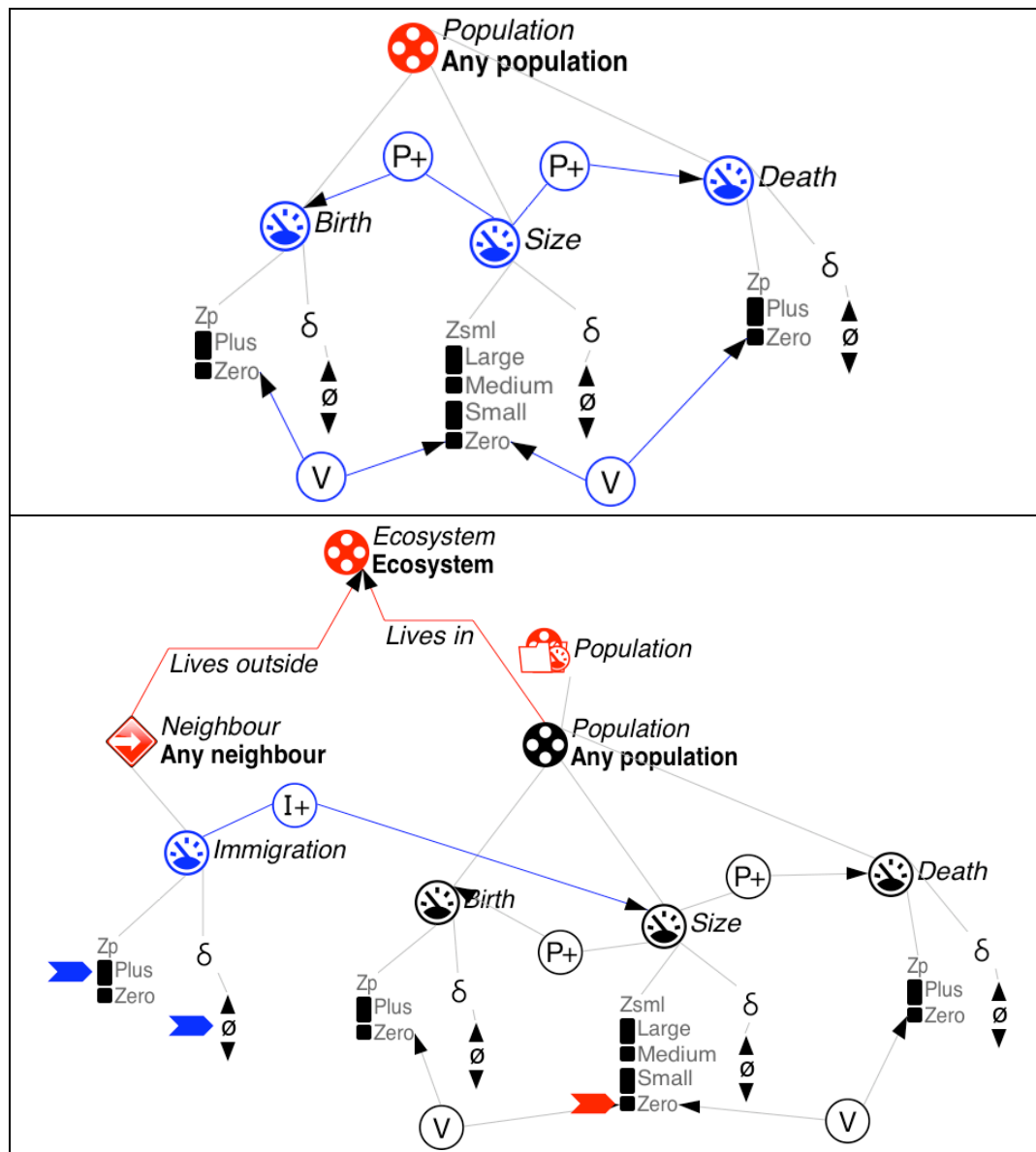


Figure 8: Model ingredient icons visualising model fragment contents. The fragment at the top shows the fragment 'population', the fragment below shows the fragment 'Colonisation', also shown in Figure 6 (RHS).

Table 4: A small selection of the Garp3 icons for model ingredients

Model fragment	Entity	Quantity	Influence (positive)	Proportionality (positive)	Value correspondence	Agent

One of the important goals behind the redesign of the model ingredients was to create sparse diagrams, in order for complex models to remain readable as much as possible. The following principles have therefore been applied:

- Model ingredients
 - Single symbol: use icon, based on QR vocabulary
 - Class name: use text label, because icons are impractical
 - Instance name: use text, as local identifier
- Binary relationships between model ingredients
 - Line and text label if user defined (e.g. structural relations)
 - Icon if ingredient belongs to QR vocabulary (e.g. equalities)
 - One default line per interactive diagram (no label needed)
- Clustered ingredients
 - Quantity spaces (magnitudes and derivatives): list vertically - from low (at bottom) to high (at top)
- Colour coding:
 - Conditions: red
 - Consequences: blue
 - Re-used model fragment: green

Similar to how action icons are used, a tool-tip text is always shown when the mouse is on top of a model ingredient icon. Moreover, in addition to showing type information, the tool-tip text will also show the 'remarks' that may have been provided by the modeller. Figure 7 shows such a 'remark' that has been provided as a description for the quantity space 'zlnhm'.

7 Toolbars

Many toolbars have been added in the new software to aid modelling. The benefit of these toolbars is twofold. First, toolbars significantly reduce the amount of work that needs to be performed in order to build and simulate a model (in terms of mouse clicks and mouse movements). Second, the toolbars are context sensitive and provide users with feedback on possible actions in a certain context. Figure 9 shows the *entity hierarchy editor* and highlights both these features. Using the toolbar, the number of steps needed to add a new entity to the hierarchy has been reduced. Instead of pulling down a menu, via *edit*, and then selecting an option in the pulled down menu, the user can now immediately select the *add new entity* icon (see also Table 3 for icon meaning).

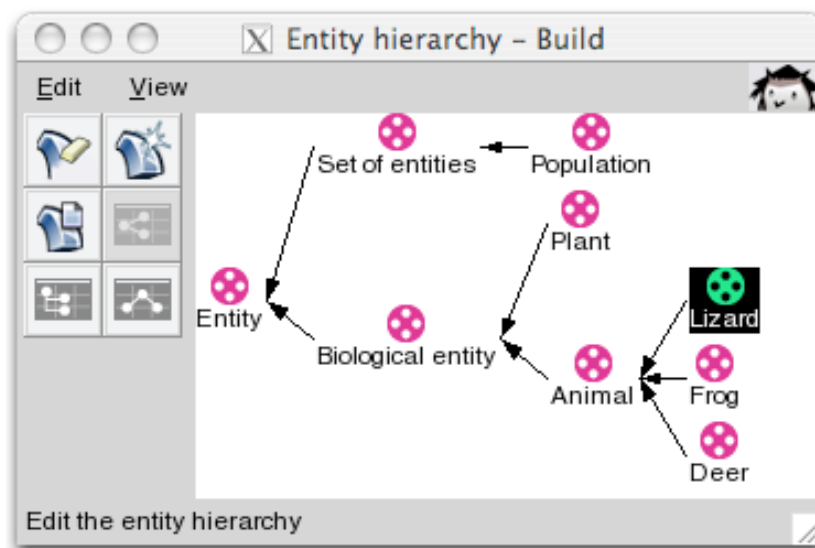


Figure 9: Toolbar and model ingredient icons in the entity editor

In Figure 9 the model ingredient *lizard* is selected. The toolbar shows the three actions that a user may now perform on the selected entity: add a new (subtype) entity, delete the selected entity, or show the properties of the selected entity. Notice that the user may also change the layout of the whole diagram (into: list or vertical).

The idea of toolbars being context sensitive is maybe even more apparent in Figure 6 (RHS). The figure shows that currently no ingredient is selected. The toolbar on the left shows the possible actions in this context (reading from top to bottom and from left to right):

- Add entity as condition
- Add entity as consequence
- Add assumption as condition
- Add agent as condition
- Add model fragment as condition

Selecting ingredients in the canvas changes the actions available in the toolbar. For instance, when an entity is selected (which by itself is not a condition, and thus not blue coloured) the available options become (reading from top to bottom and from left to right, although not explicitly shown in Figure 6, RHS):

- Add attribute as condition
- Add attribute as consequence
- Add quantity as condition

- Add quantity as consequence
- Add assumption as condition

Toolbars are also available in the simulate part of the new software, as for instance shown in Figure 10 (main screen) and Figure 11 (dependency screen). The toolbar in the main screen is also context sensitive. In the previous version of the software an error message was given when impossible actions were selected. Using a context sensitive toolbar prevents this by automatically supporting the user in selecting *possible* options. Table 5, and the text below, provide an overview of the meaning of the main screen icons and the accompanying tool-tip text.

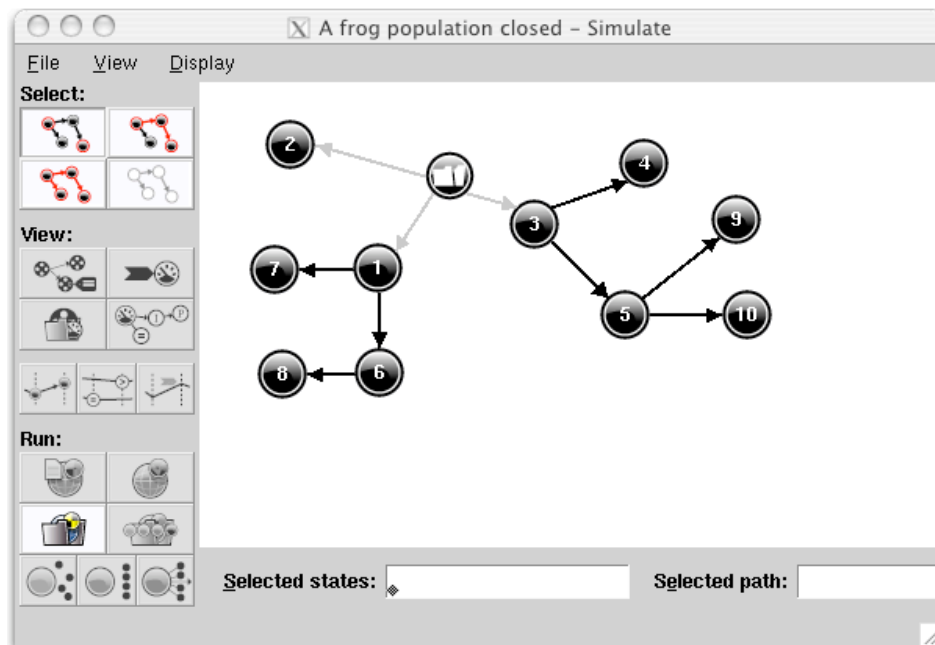


Figure 10: Toolbar for main screen simulate

Table 5: Garp3 interface – Toolbar main screen simulate

Sel1		Sel2	
Sel3		Sel4	
View1		View2	
View3		View4	
Hist1	Hist2		Hist3
Sim1		Sim2	
Sim3		Sim4	
Sim5	Sim6		Sim7

There are four options for selecting states in the state graph and they have the following tool-tip text:

- Sel1: Select individual states
- Sel2: Select a path
- Sel3: Select all states
- Sel4: Deselect all states

There are four views to investigate specific details in the state graph and they have the following tool-tip text:

- View1: Show entities, configurations and attributes (for selected states)
- View2: Show quantity values (for selected states)

- View3: List model fragments (for selected states)
- View4: Show dependencies (for selected states)

There are three ways to inspect histories over a set of states and they have the following tool-tip text:

- Hist1: Transition history
- Hist2: Equation history
- Hist3: Value history

There are seven options to run a simulation and they have the following tool-tip text:

- Sim1: Open trace window
- Sim2: Simulation preferences
- Sim3: Select a scenario to simulate
- Sim4: Full simulation of current scenario
- Sim5: Terminate selected states
- Sim6: Order selected states
- Sim7: Find successors for selected states

In the state graph in Figure 10 no states are selected, hence none of the view and history options are available. In contrast, the selection options are always available, because they are independent of specific states being selected. This is also the case with *select a scenario to simulate* (Sim3), because it is always possible to select a new scenario and run its simulation.

Figure 11 shows the *dependency view* and its toolbar (on the left). Different from the toolbars discussed above, this toolbar is not context sensitive. Instead, it implements a kind of toggle menu and provides means to show (or not show) certain dependency types in the canvas.

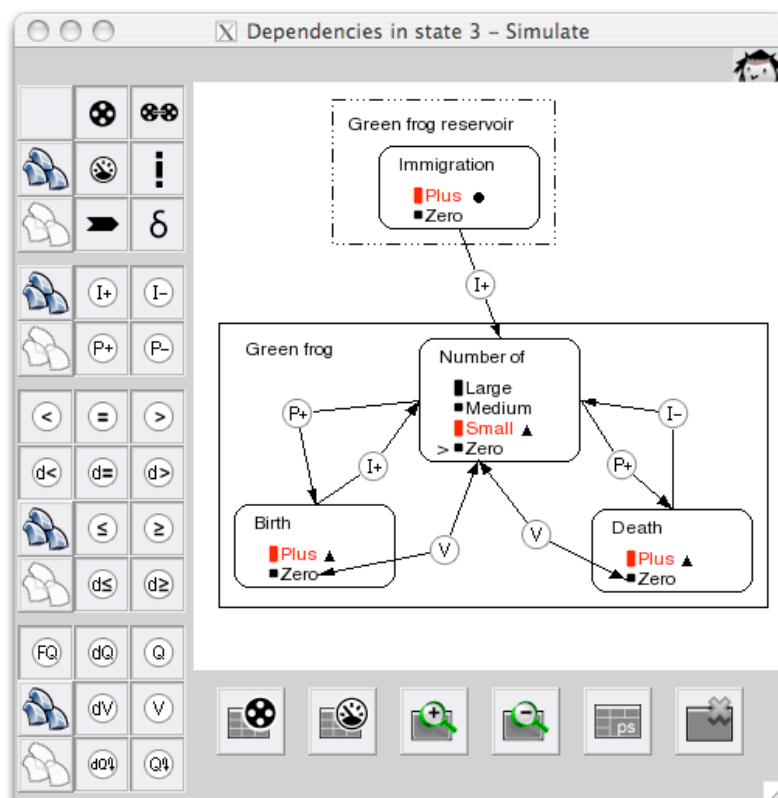


Figure 11: Toolbar for dependency screen in build

Compared to the original dependency screen the following improvements have been made. The options have been regrouped into four meaningful units (instead of two): structure and quantities, causal dependencies, (in)equalities, and correspondences. Each group has been provided a 'select all' and 'deselect all' button. There are three new types of correspondences. The functionality for displaying structural details and quantities has been debugged and optimised.

8 Equation and value history

The *quantity value history view* (or *value history*) shows an overview of all values of the selected quantities for the different (selected) states in the simulation, as shown in the front window in Figure 12. To include the value history in Garp3, new icons have been introduced for the *sort* feature (to sort the quantities by quantity name or their entity's name), the *select* feature (*all* or *none*), and the *graph* feature (*draw* a graph, and *erase*).

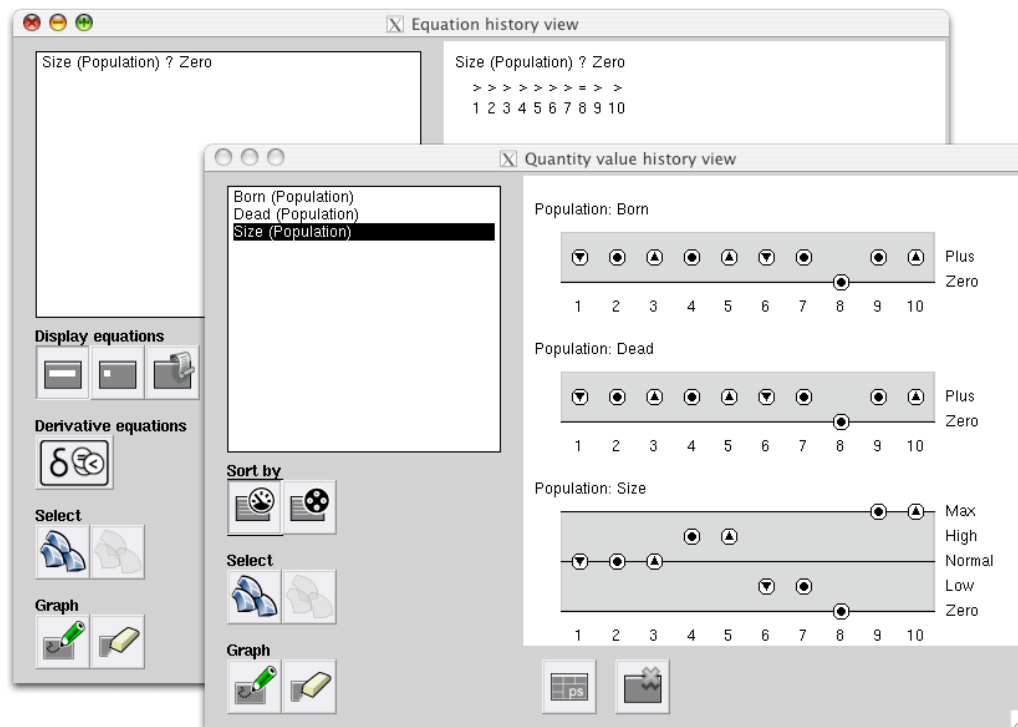


Figure 12: Equation (picture in the back) and value history (picture in the front)

Furthermore, a new view has been introduced, called the *equation history* (see the window in the background in Figure 12). In this variant of the *value history*, an overview is given of ordinal relationships for the different (selected) states in a simulation. This makes it easy to see which ordinal relationships apply to which states, as well as how these change over states.

On the top left side of this window, a list of generic equations between quantities is shown, where the (in)equality sign ($<$, $<=$, $=$, $>=$, or $>$) is replaced by a '?'. This list indicates that information is available about the ordinal relationships between the quantities (and values) involved. The equation may consist of a quantity, a sign, and a particular value (i.e., $Q1 ? V$), or of a quantity, a symbol, and another quantity (i.e., $Q1 ? Q2$). When a particular equation is selected (i.e., $Size(Population) ? Zero$), and the *graph* button is pressed, a table is drawn on the right side of the *equation history view*. In this table, the following visualisation is used:

- The selected states are shown on the bottom, along the X-axis.
- The possible (in)equality types are shown on the right, along the Y-axis.
- The (in)equality in a particular state is shown as a symbol which should be inserted in the place of the '?'.

In order to keep the visualisation compact, the generic equation is used as a header for the table, while the table itself only shows the symbol above the state numbers.

There are three options for the list of possible equations to be visualised, activated by the three *display equations* buttons:

- Show long names: display quantities in the format QuantityName(EntityName). This is the default option.
- Show short names: display quantities in a format without the entity names.
- Show names in legacy mode: display quantity names in legacy mode.

This allows the user to decide about the trade-off between the clarity of added detail and more efficient use of space.

A further option button (*include derivative equations*) is available to include equations between derivatives or not. By default, this option is turned off, because these equations are usually of lesser importance than the equations between quantities (and values).

9 Simulate and build interoperability

The integration of the three tools Garp, VisiGarp, and HOMER creates interoperability between the *build* and *simulate* environment in Garp3. This interoperability greatly speeds up the interaction process because it facilitates the move from building and editing a model to simulating it. This is mainly due to the combination of build and simulate functionalities in the Garp3 main screen, but there are also several direct links from within the workbench: e.g. from the Scenarios – Build screen, which is used to select a scenario to edit, the user can also choose to simulate the selected scenario.

It is also possible to move from the *simulate* environment back to the *build* environment, directly. When running a simulation, the simulation results may point to sub-optimalities in the model, or the need may occur to experiment with slight variations of the model. Therefore, direct links have been included which allow the user to move back from specific simulation results directly to the associated editor in the *build* environment. An example is presented in Figure 13: in the *simulate* view for inspecting quantity values in a particular state, there is a button *edit selected quantity*, which leads back to the *quantity definitions* in the *build* environment. Similar direct links are implemented from the *state-transition graph* view menu (in simulate) to the *scenario editor* for the current scenario (in *build*), and from the *list model fragments* (in *simulate*) to the *model fragment editor* (in *build*).

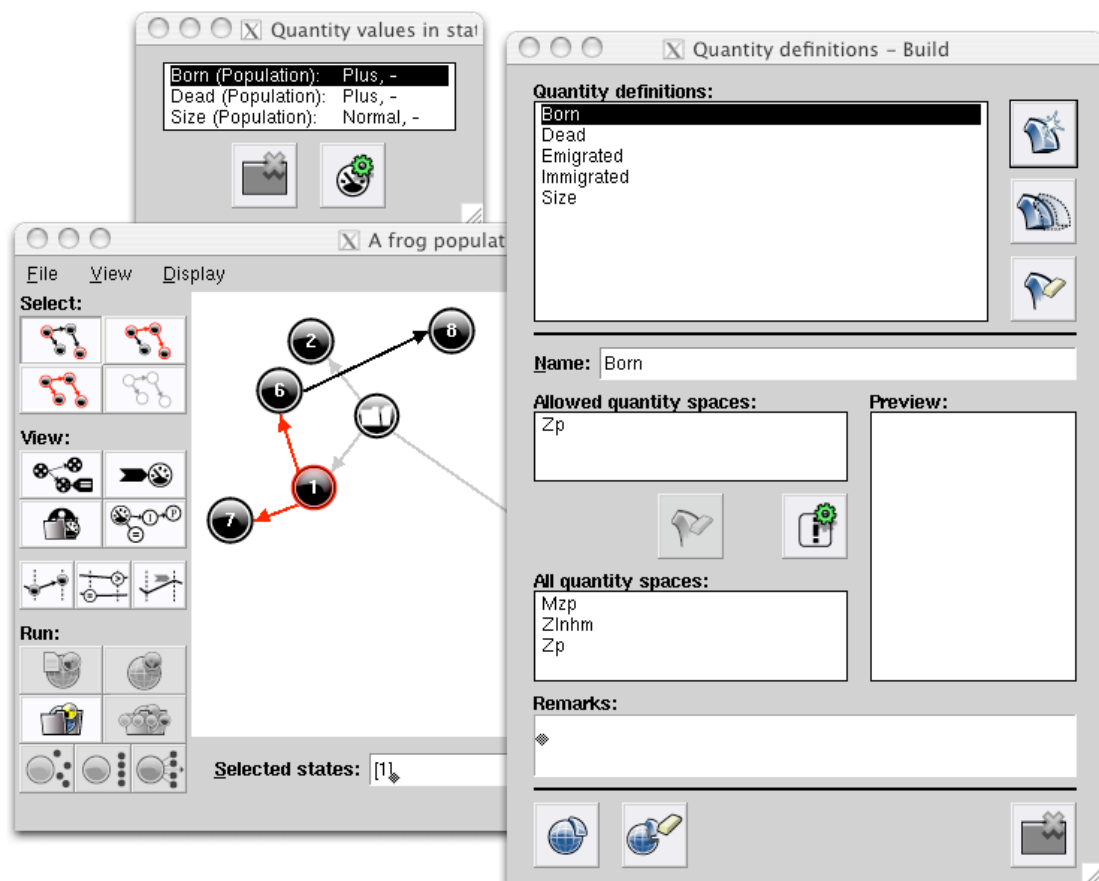


Figure 13: Simulate and build interoperability

10 Saved simulations

A new feature in Garp3 is the ability to save simulation results into the model (see Figure 14). This allows the user to store particular simulation results (i.e., the first simulation run of a particular scenario, with 10 states), to be accessed at a later time without the need to search for the right scenario and run the simulation again. This feature is especially useful when there are multiple scenarios, or when the resulting simulations are so large that not all states can be investigated at once.

Another intended use of this feature is to support debugging of models and beta releases of the software. A model with a saved simulation that contains a suspected bug can be included into a bug report, which greatly facilitates reconstructing the exact circumstances in which the anomaly occurred.

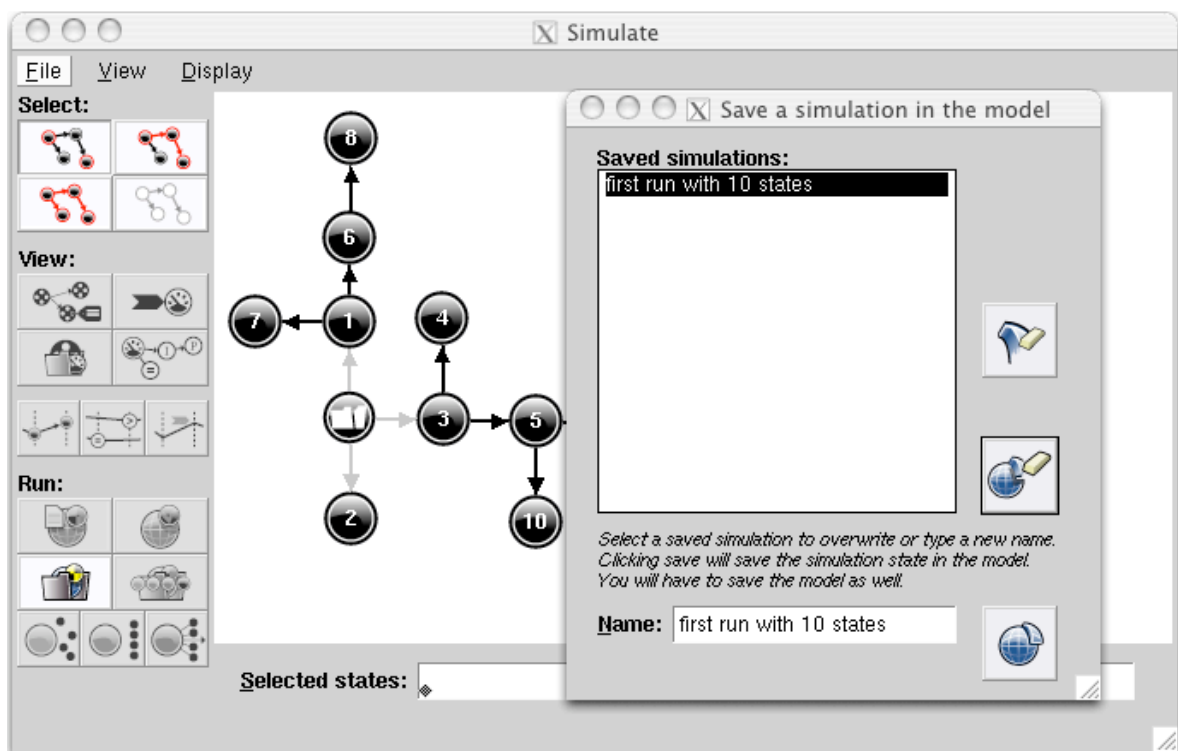


Figure 14: Saving simulation results into the model

11 Additional features

Figure 15 shows the Garp3 main screen following the phase 3 software development period.

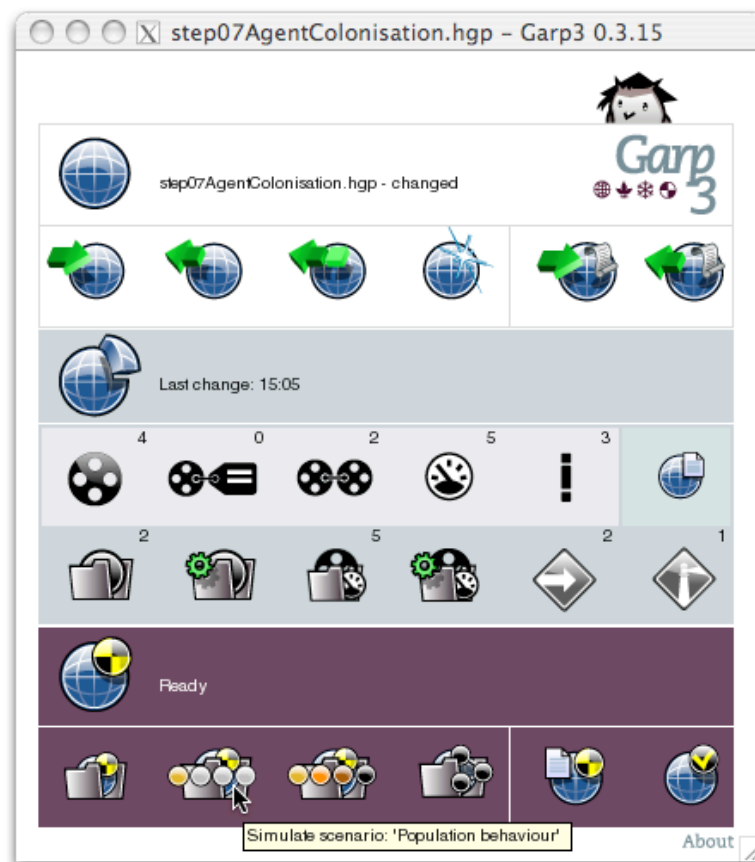


Figure 15: Main screen of the Garp3 workbench, following phase 3 (with tool-tip text for the action button 'Run current simulation')

Compared to the version resulting from phase 2, the following additional user support and control options are included:

- **Help button.** Each interactive screen in the workbench has a unique reference to a webpage at the QRM portal <http://hcs.science.uva.nl/QRM/>. Within task 4.4 this will be used to create an interactive and context sensitive online help system.
- **Conditional model fragment refinement.** Conditional model fragments can be refined into their subtypes, when the fragment in which they are conditions is refined.
- **Default functions,** e.g. generate all values for Q1. This feature is particularly meant to provide different characteristics to exogenous quantities (quantities external to the system). During the simulation a quantity will show the behaviour according to the default function, when that behaviour is assigned to the quantity in the scenario. The following default behaviours can be specified for exogenous quantities:
 - Constant
 - Generate all values
 - Exogenous: sinus, free, plus, steady, or min
- **Model fragment overview as diagram** (already shown in Figure 6 LHS), including:
 - Make selected model fragment inactive/active (on/off switch)
 - Show default view

- Save current view
 - Open another view
 - Show parent-child relations (on in Figure 6, LHS)
 - Show conditional relations (off in Figure 6, LHS).
- *Simulation preferences* (see next section).
- *Engine trace* formatted and made available (see next section).
- *Save diagrams as EPS*. Export function to save screen diagrams as EPS documents (in order to support printed documentation of model content and simulation results).
- *Meta data*. Information can be specified concerning: title, author, contributors, contact email, subject/keywords, model version, domain, model goals, intended audience, known model limitations, language, bibliographic citation, bug-reports, etc. The idea is to augment the model with additional information in order to enable debug-support (see task 4.3 in the NaturNet-Redime project) and support model reuse.
- *Technical documentation*.

12 Trace options and Simulation preferences

The trace window and the simulation preferences are simultaneously shown in Figure 16, although they are two independent features of the workbench. There are 11 trace options (such as: show search for applicable model fragments, show in/equality reasoning details, show influence resolution, etc.) that can support the modeller in finding out details about the working of the engine and the appropriate representation of model ingredients. The output of the tracer is currently not optimised for non-AI/QR users, and presents details that may be difficult to understand. Further formatting and improving the output of the tracer is part of ongoing work.

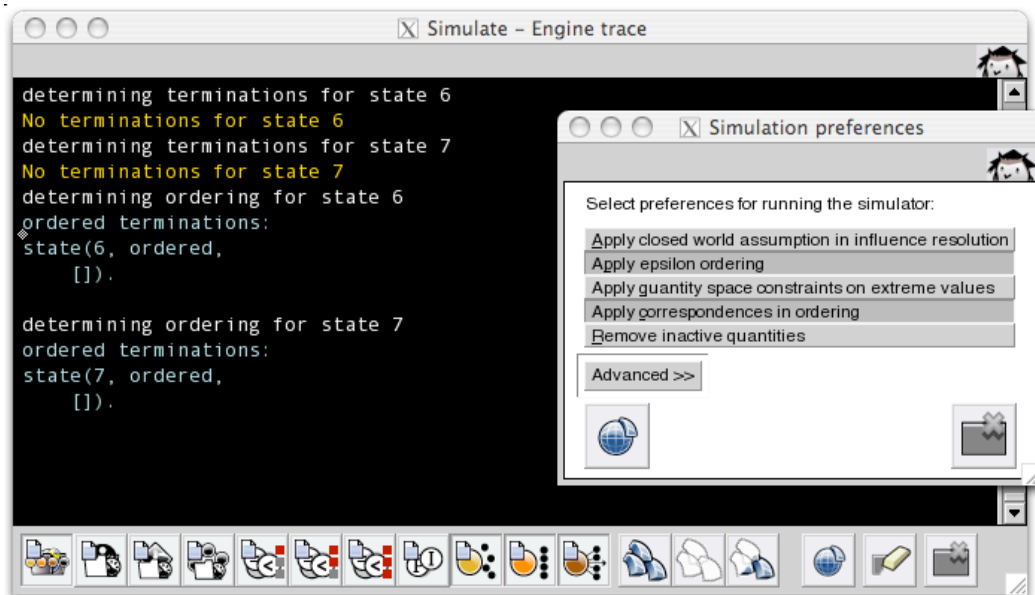


Figure 16: Trace window (in the back) and simulation preferences (in the front)

Discussions with experts inspired us to augment the qualitative reasoning engine with user definable features controlling the problem solving behaviour of the engine (Bredeweg and Salles, 2005). These simulation preferences allow modellers to build models taking a certain perspective. In total 19 simulation preferences can be turned on or off; table 6 enumerates the ‘easy’ ones shown in Figure 16.

Table 6. Some simulation preferences in Garp3 that users can change

Type	Description	Default
Apply closed-world assumption in influence resolution	Assume that the impact of directly (or indirectly) influencing quantities is zero when their magnitude (or derivative) cannot be determined.	off
Apply epsilon ordering	Changes from a point (and equality) happen instantly and therefore precede changes that take at least some small amount of time.	on
Apply quantity constraints on extreme values	Quantities may not increase in the highest point (or decrease in the lowest point) of their quantity space.	off
Apply correspondence in ordering	Have corresponding quantities and quantity values change simultaneously.	on
Remove inactive quantities	When processes stop, related quantities (mainly rates) may become superfluous. Remove these quantities.	off

Probably the most obvious example is simulating with or without applying a closed world assumption in influence resolution. When the assumption is active the main impact is as follows: unknown values of directly influencing quantities are set to zero, and unknown derivatives of indirectly influencing quantities are set to zero. In other words, the influence of unknown phenomena is defined as being zero (which is different from unknown).

13 Conclusions

This deliverable has presented a summary of the main features of Garp3, a workbench for building, running, and inspecting qualitative models. The workbench offers an easy access to high-end qualitative simulation software, providing non-AI/QR experts with the possibility to use Qualitative Reasoning technology without having to understand low-level implementation details of such automated reasoners. The realisation of Garp3 is part of a bigger effort to support users in actually using Qualitative Reasoning technology to develop and organise their understanding of system behaviour, including the development of a *Curriculum for learning about QR* (Bredeweg et al., 2006) and a *Structured approach to modelling* (Bredeweg et al., 2005). The software and additional information can be downloaded from <http://hcs.science.uva.nl/QRML/>.

The software results of phase two have been delivered in September 2005 and have been used during the Amsterdam workshop (October 11-14th 2005) without any serious technical problems. During the third phase additional features have been implemented that further improve the usability (e.g. trace options) and user-control (e.g. simulation preferences). The software resulting from this phase has been used during the Sofia workshop (March 27-31st, 2006), also without any serious technical problems. This is encouraging for the expected functioning and stability of the final release of the Garp3 workbench.

Future work will focus on supporting collaborative modelling and model reuse, and will include a repository for uploading, indexing, and downloading models and model parts, as well as means for modellers to copy/paste model ingredients between models. Automatically preserving model consistency and supporting model debugging are essential for this, and need further development.

14 Bibliography

- Bessa Machado, V. (2004) Supporting the Construction of Qualitative Knowledge Models, Ph.D. thesis, University of Amsterdam, Amsterdam, The Netherlands.
- Bouwer, A. (2005) Explaining Behaviour - Using Qualitative Simulation in Interactive Learning Environments, Ph.D. thesis, University of Amsterdam, Amsterdam, The Netherlands.
- Bredeweg, B., Liem J., Bouwer, A., and Salles, P. (2006). *Curriculum for learning about QR modelling, Naturnet-Redime*, STREP project co-funded by the European Commission within the Sixth Framework Programme (2002-2006), Project no. 004074, Project Deliverable Report D6.9.1.
- Bredeweg, B. and Salles, P. (2005) The Ants' Garden: Complex interactions between populations and the scalability of qualitative models, *AI Communications*, volume 18, issue 4, pages 305-317.
- Bredeweg, B., Salles, P., Bouwer, A., and Liem J. (2005) *Framework for conceptual QR description of case studies*, Naturnet-Redime, STREP project co-funded by the European Commission within the Sixth Framework Programme (2002-2006), Project no. 004074, Project Deliverable Report D6.1.
- Bredeweg, B. (1992) Expertise in Qualitative Prediction of Behaviour. Ph.D. thesis, University of Amsterdam, Amsterdam, The Netherlands.
- Groen, R. (2003) Supporting Model Building, Master thesis, University of Amsterdam, Amsterdam, The Netherlands.
- Haverkort, N. (2000) Graphical Representation of Qualitative Simulations, Master thesis, University of Amsterdam, Amsterdam, The Netherlands.
- Jellema, J. (2000) Ontwerpen voor Ondersteuning - De rol van taakkennis bij ondersteuningsontwerp, Master thesis, University of Amsterdam, Amsterdam, The Netherlands (in Dutch).
- Linnebank, F. (2004) Common Sense Reasoning - Towards Mature Qualitative Reasoning Engines, Master thesis, University of Amsterdam, Amsterdam, The Netherlands.
- Tjaris, P. (2003) Kwalitatieve simulaties als middel tot kennisoverdracht, Master thesis, University of Amsterdam, Amsterdam, The Netherlands (in Dutch).
- Werf, v.d. R. (2003) The Use of Qualitative Modelling and Simulation Tools in High School Education: an Engineering Study, Master thesis, University of Amsterdam, Amsterdam, The Netherlands.