Garp3 – A new Workbench for Qualitative Reasoning and Modelling

Bert Bredeweg[1], Anders Bouwer[1], Jelmer Jellema[2], Dirk Bertels[1, 3], Floris Linnebank[1], and Jochem Liem[1]

[1]Human Computer Studies, University of Amsterdam
Kruislaan 419 (matrix I), 1098 VA Amsterdam, The Netherlands, E-mail: {bredeweg,bouwer,jliem}@science.uva.nl
[2]Spin in het Web, Apeldoorn, The Netherlands, E-mail: {Jelmer@spininhetweb.nl}
[3]Louter Vormgeving, Amsterdam, The Netherlands, E-mail: {dirk@louter.info}

Abstract

Easy to use workbenches for Qualitative Reasoning and Modelling are virtually nonexistent. This has a limiting effect on the use and update of the technology by a larger audience. We present Garp3, a user-friendly workbench that allows modellers to build, simulate, and inspect qualitative models. Garp3 can be used by experts to create conceptual models in situations were numerical information is sparse or unavailable. Garp3 can also be used in educational settings or dissemination activities to illustrate and educate typical features of Qualitative Reasoning.

Introduction

Qualitative Reasoning (QR) provides means 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. Qualitative models provide formal means to externalise thought on such conceptual notions. There is growing interest by ecological experts to create qualitative models of phenomena for which numerical information is sparse or missing[1]. There is also an interest to capture and simulate conceptual knowledge as such (cf., Jørgensen and Bendoricchio, 2001). However, building qualitative models is difficult and hampered by the lack of easy to use tools.

Usable workbenches for Qualitative Reasoning and Modelling are almost nonexistent. 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 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 is the rich and detailed semantics used, which are based on QR formalisms. However, 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 software. Although this is effective, it has the obvious drawback of not using the full potential of QR and the means it provides for representing conceptual knowledge.

In our approach we want to preserve the full expressiveness of the QR formalism. Moreover, we want to address experts and support them in articulating and capturing conceptual knowledge. We have therefore developed Garp3, a user-friendly workbench that allows modellers to build, simulate, and inspect qualitative models. The software uses a diagrammatic approach for representing model content, and graphical buttons to communicate the available user options and manipulations. This paper discusses this workbench, focussing on the user interface and how modellers can use it. This paper does not particularly discuss the QR engine, which is part of the workbench, but see Linnebank (2004) for the latest details.

Garp3 – Background and Main features

Garp3 is implemented in SWI-Prolog[2] and based on previous software, including: Garp2 for simulating models (Linnebank, 2004), Homer for building models (Jellema, 2000), and VisiGarp for inspecting simulation results (Bouwer, 2005). Integrating these three tools has led to one new tool that incorporates all of the original functionalities, and thus incorporates the advantages of each tool, but also adds interoperability and an easy to use uniform user interface. Figure 1 shows the main screen of the new Garp3 workbench.

The following main features have been realised in the new workbench (in the next sections a selection of these is discussed in more detail):

- Organising all functionalities in an integrated workbench, and provide access to the key functions from the main screen.
- Seamless interoperability between Build and Simulate context (e.g., being able to jump from a model fragment present in a state in Simulate to that same model fragment in the Build context to make changes).
- Uniform use of vocabulary, graphical icons, diagrams, and colour.

[1] See e.g. www.naturnet.org & http://www.isei5-conference.elsevier.com/

[2] http://www.swi-prolog.org/

- Diagrammatic visualisation of model content where possible and logical (e.g., in scenario and model fragment editors in the Build context, and in dependency screen in Simulate context).
- Usage of toolbars to shorten user actions as much as possible (most options can also be accessed via menus and 'in screen mouse-selection').
- Scenario manipulation in simulation mode (e.g., setting initial quantity values).
- Regular import/open and export/save of files storing models.
- Simple copy/paste within a model (e.g., scenarios and model fragments).
- Context sensitive online help (the functionality is implemented in the workbench, but the corresponding websites are currently under construction).
- The option to save simulations in the model-file (e.g., for easy access later on, and for debug support).
- Tool-tip (standard: text explaining action buttons, and special for QR models: 'description texts' for model ingredients as provided by modellers while modelling, for instance in scenario and model fragment editors).
- Diagrammatic overview of model fragments (showing sub-type hierarchy and/or conditional relationships), the ability to save multiple user-made screen layouts and the ability to turn model fragments on and off (an essential feature during modelling, particularly debugging).
- Simulation preferences (e.g., simulate with 'closed-world' assumption on or off, etc.).
- Formatted engine trace, with the option to select the kind of inferences that should be shown.
- Export function to save screen diagrams as EPS documents (in order to support printed documentation of model content and simulation results).

Figure 1: Main screen of the Garp3 workbench
(with tool-tip text for the action button 'Run current simulation')

Action buttons

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

As the vocabulary used in qualitative reasoning and modelling is in principle unknown to non-QR/AI scientists, it is impossible to develop a set of action buttons that is immediately understood by such users. By definition, the vocabulary and the corresponding buttons have to be learned by the potential users. Therefore, more important than immediately 'understanding' the meaning of a button, 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 buttons 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: *filing* 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 Garp3 workbench is divided into four areas: one for each of the groups of manipulations mentioned in Table 1, and one for additional options such as 'help' and 'about'. For each group of manipulations additional buttons have been developed. Table 2 shows the manipulations that are accessible from the main screen. This table compares to the screenshot of the Garp3 main screen shown in Figure 1. To benefit from graphics and text, a tool-tip text is always shown when the mouse is on top of an action button. The tool-tip text names the user-action that is accessible via the button.

Table 2: Garp3 interface – Main screen action buttons

					Help
File icon	*Model status info*				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:

- 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

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 while modelling. 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 (model and maker meta-data)
- B7: Open scenarios editor
- B8: Edit scenario: <name last changed scenario>
- B9: Open model fragments editor
- B10: Edit model fragment: <name last changed FM>
- B11: Open agents hierarchy editor
- B12: Open assumptions hierarchy editor

There are six action icons related to simulating a model. S2 and S3 provide shortcuts 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 scenario: <name last simulated scenario>
- S3: Full simulation scenario: <name last simulated scenario>
- 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 (Owl): opens default browser with help on QRM
- Logo: opens browser with general Garp3 information
- About: provides information about the developers

Following the initial icons for 'model' ('blue earth circle') and 'model ingredient' (a part from the 'blue earth circle'), 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.

Table 3: Model and model ingredient action buttons (from left to right: Save ingredient in model, Delete ingredient from model, Create new ingredient, Show ingredient properties, Copy ingredient, and Delete ingredient)

Figure 2 (LHS) shows the selector for model fragments and illustrates how these icons are used in the context of a specific model building activity. 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
- Make selected model fragment inactive/active
- Show default view
- Save current view
- Open another view
- Show parent-child relations (on in Figure 2, LHS)
- Show conditional relations (off in Figure 2, LHS)

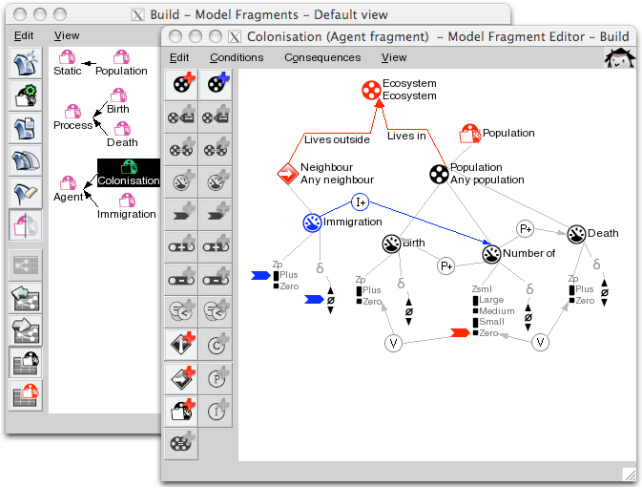


Figure 2: Action buttons in the model fragment selector (LHS). From this selector, model fragments can be opened for editing as is done for Colonisation (RHS).

Notice that the copy options for model fragments and scenarios 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.

Visualising model ingredients

The graphical icons used to signify the model ingredients have been redesigned in the new software. Figure 3 and 4 show the content of the model fragments 'Population' and 'Colonisation'. The meaning of most of the icons used in these figures is shown in Table 4.

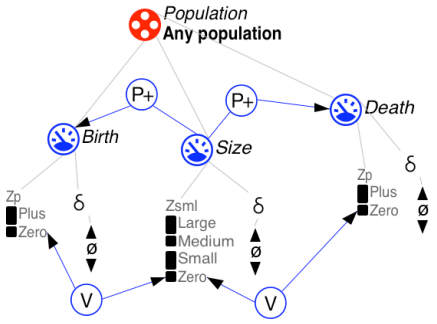


Figure 3: Ingredient icons visualising the fragment 'Population'.

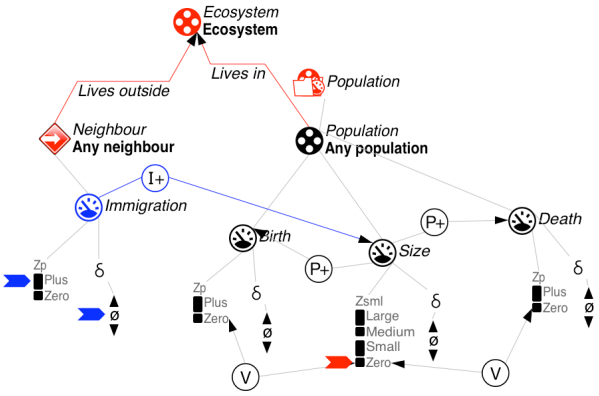


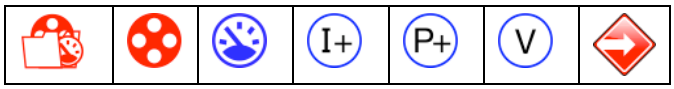
Figure 4: Ingredient icons visualising the model fragment 'Colonisation', also shown in Figure 2 (RHS).

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 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 (grey/black) per interactive diagram (no label needed, e.g., 'belongs to')
- 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 buttons 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.

Table 4: A small selection of the Garp3 icons for model ingredients (from left to right: Model fragment, Entity, Quantity, Influence, Proportionality, Value correspondence, and Agent)



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). For instance, instead of pulling down a menu, via 'Edit', and then selecting an option from the pull-down menu, the user can now immediately select the 'add new entity' icon (see also Table 3 for icon meaning). Second, the toolbars are context sensitive and provide users with feedback on possible actions in a certain context. Figure 5 shows the entity hierarchy editor and highlights both these features.

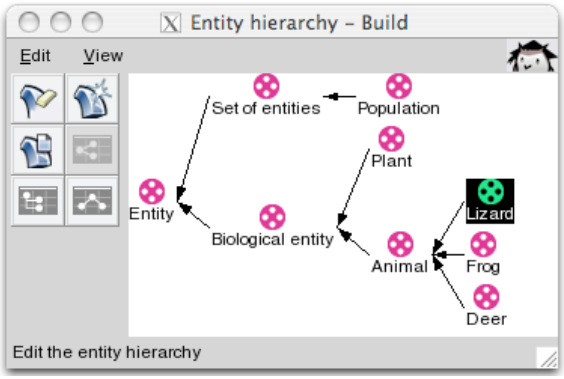


Figure 5: Toolbar and model ingredient icons in the entity hierarchy editor

In Figure 5 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 2 (RHS). The figure shows a model fragment editor in which 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 will change the actions available in the toolbar. For instance, when an entity is selected the available options will become (reading from top to bottom, and from left to right, although not explicitly shown in Figure 2):

- 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 workbench, as for instance shown in Figure 6 (main screen) and Figure 7 (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 provides an overview of the meaning of the main screen toolbar buttons. There are four options for selecting states in the state graph, four views to investigate specific details in the state graph, three ways to inspect histories over a set of states, and seven options to run a (partial) simulation.

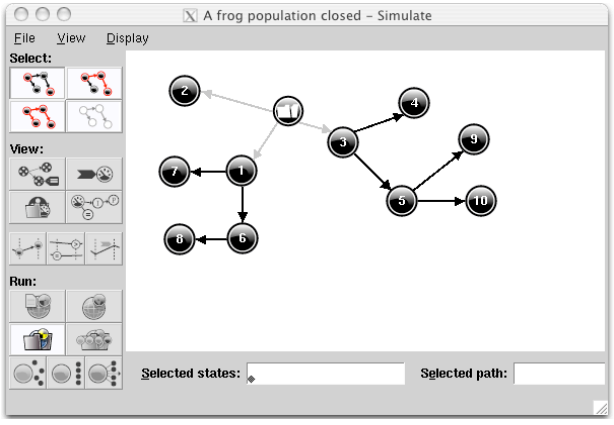


Figure 6: Main screen Simulate with toolbar

There are no states selected in the state graph shown in Figure 6, 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', because it is always possible to select a new scenario and run its simulation.

Table 5: Garp3 interface – Toolbar main screen Simulate

Select individual states		Select a path
Select all states		Deselect all states
Show entities, configurations & attributes (for selected states)		Show quantity values
List model fragments		Show dependencies
Transition history	Equation history	Value history
Open trace window		Simulation preferences
Select a scenario to simulate		Full simulation current scenario
Terminate selected states	Order selected states	Find successors for selected states

Figure 7 shows the dependency view and its toolbar. 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 model ingredients in the canvas. Compared to the original dependency screen (VisiGarp, Bouwer, 2005) 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 given a 'select all' and 'deselect all' button. There are three new types of correspondences (e.g. derivative correspondence). The display functionality for structural details and quantities has been debugged and optimised.

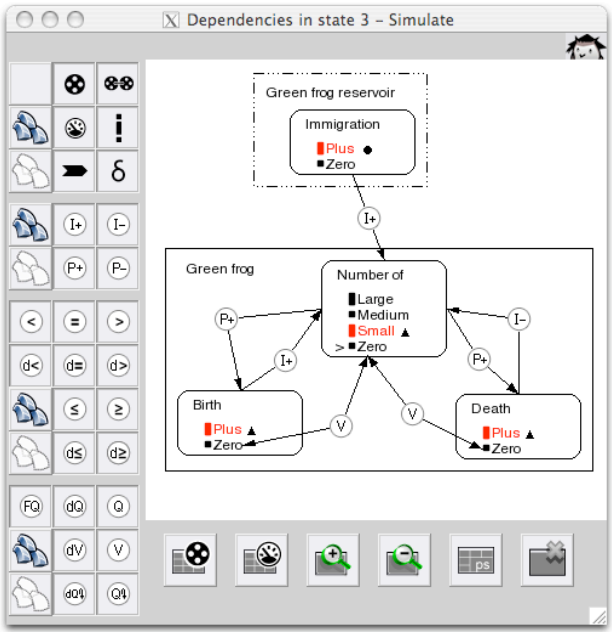


Figure 7: Dependency screen with toolbar in Simulate

Equation and value history

The value history (or quantity value history view) shows an overview of all values for the selected quantities for the selected states in the simulation, as shown in the front window in Figure 8. In the value history window of 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*).

Furthermore, a new view has been introduced, called the equation history (see the window in the back in Figure 8). 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) enumerated. 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.

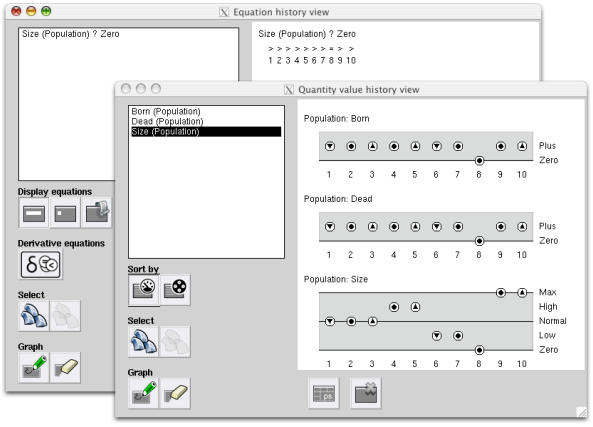


Figure 8: Equation (in the back) and value history (in the front)

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).

Simulate and Build interoperability

The seamless integration of the old tools (Garp2, 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.

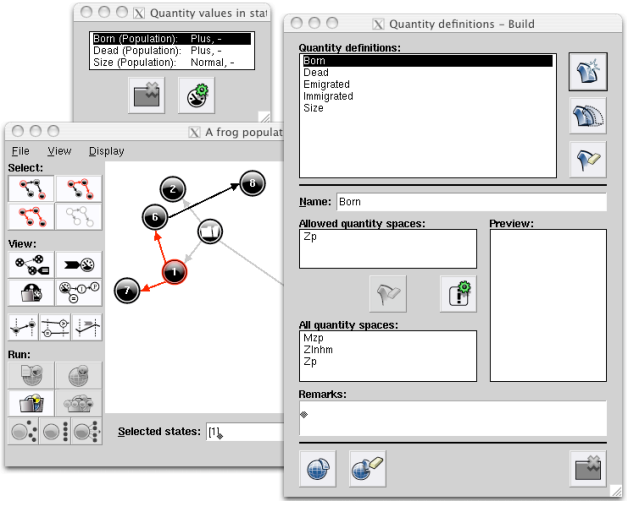


Figure 9: Simulate and Build interoperability

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 9: in the *simulate view* for inspecting quantity values in a particular state (state 1 in Figure 9), there is a button *edit selected quantity* (LHS top, in Figure 9), which leads back to the *quantity definitions* in the Build environment (RHS, in Figure 9). 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). This kind of functionality greatly improves the usability of the workbench for modelling activities.

Saved simulations

A new feature in Garp3 is the ability to save simulation results into the model (see Figure 10). 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.

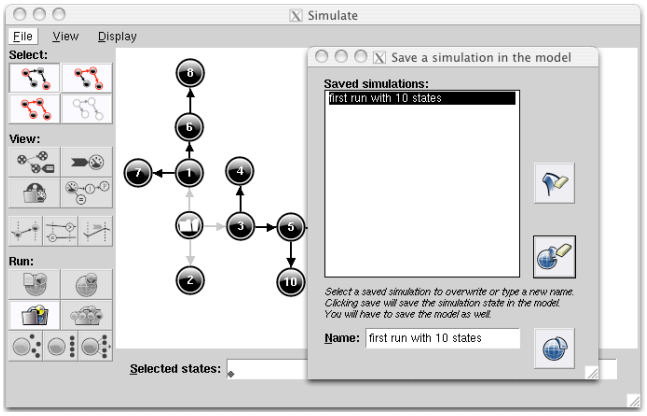


Figure 10: Saving simulation results into the model

Trace options and Simulation preferences

The trace window and the simulation preferences are simultaneously shown in Figure 11, 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.

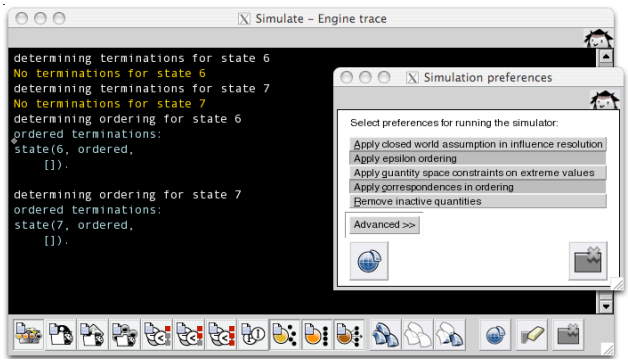


Figure 11: Trace window (in the back) and preferences (in front)

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. 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). In total 19 simulation preferences can be turned on or off; table 6 enumerates 'easy' ones shown in Figure 11.

Table 6. 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) 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

Exogenous Quantities

Garp3 implements new mechanisms for modelling quantities that exhibit exogenously defined behaviours. Exogenous quantities are those that may influence the system behaviour but are not influenced by quantities representing the system (Rickel and Porter, 1997). Seven types of mechanisms for handling exogenous quantities are available and can be assigned to quantities in the scenario:

- Generate all values: the engine tries to generate all possible magnitudes in each state.
- Constant: the magnitude and/or derivative remain as defined in the scenario.
- Increasing: the derivative value is kept positive.
- Decreasing: the derivative value is kept negative.
- Steady: the derivative value is kept zero.
- Sinusoidal: the derivative is positive until the maximum magnitude is reached. Then the derivative changes to zero, and starts moving in the opposite direction until the minimum magnitude is reached, etc.
- Random: similar to sinusoidal, but instead of continuously moving towards the extreme values, random can assume any derivative value (albeit obeying the continuity law) and move in any direction.

Conclusion

This paper has presented 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 QR 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, and includes 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 can be downloaded from http://hcs.science.uva.nl/QRM/. Domain experts within the NaturNet-Redime project are currently using the workbench to capture knowledge about issues relevant to sustainable development.

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.

Acknowledgement

The research presented in the paper is co-funded by the European Commission within the Sixth Framework Programme for Research and Development (2002-2006) (project NaturNet-Redime, number 004074, www.naturnet.org). We want to thank the anonymous reviewers and the participants of the NaturNet-Redime workshop in Amsterdam (October 11-14, 2005) for their feedback and suggestions concerning the Garp3 workbench.

References

Biswas, G., Schwartz, D., Bransford, J. & The Teachable Agents Group at Vanderbilt. (2001) Technology Support for Complex Problem Solving: From SAD Environments to AI. In K. Forbus and P. Feltovich (Eds.). Smart Machines in Education. AAAI Press/MIT Press, Menlo Park California, USA.

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 EC within the 6th Framework Programme (2002-2006), Project no. 004074, Project Deliverable D6.9.1.

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 EC within the 6th Framework Programme (2002-2006), Project no. 004074, Project Deliverable D6.1.

Bredeweg, B. and Salles, P. (2005) The Ants' Garden: Complex interactions between populations and the scalability of qualitative models, *AI Communications*, 18(4):305-317.

Bredeweg, B. and K. Forbus, K. (2003) Qualitative Modeling in Education. *AI Magazine*, 24(4):35-46.

Forbus, K.D, Carney, K., Harris, R. and Sherin, B.L. 2001. *A qualitative modeling environment for middle-school students: A progress report*. In: G. Biswas (Ed.), The 15th International Workshop on Qualitative Reasoning, pages 65-72, St. Mary's University, San Antonio, Texas.

Forbus, K.D. (1984) Qualitative process theory. *Artificial Intelligence*, 24(1-3):85-168.

Jellema, J. (2000) *Ontwerpen voor Ondersteuning - De rol van taakkennis bij ondersteuningsontwerp*, Master thesis, University of Amsterdam, Amsterdam, The Netherlands (in Dutch).

Jørgensen, S.E. and Bendoricchio, G. (2001) Fundamentals of Ecological Modelling (3rd edition). Elsevier, Oxford.

Linnebank, F. (2004) *Common Sense Reasoning - Towards Mature Qualitative Reasoning Engines*, Master thesis, University of Amsterdam, Amsterdam, The Netherlands.

Novak, J.D. and Gowin, D.B. (1984) *Learning how to learn*. Cambridge University Press, New York, New York.

Rickel, J., and Porter, B. (1997) Automated Modeling of Complex Systems to Answer Prediction Questions. *Artificial Intelligence*, 93:201-260.