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**

D6.1
Framework for conceptual QR description of case studies

Due date of deliverable: 31/08/2005
Actual submission date: 25/10/2005

Start date of project: **1st March 2005** Duration: **30 months**

Organisation name of lead contractor for this deliverable:
University of Amsterdam[1]

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, Paulo Salles, Anders Bouwer and Jochem Liem

Abstract

This document presents a methodology that structures and supports the capture of conceptual knowledge about sustainable development using a qualitative approach. The framework defines a protocol for describing content (knowledge and expertise) that supports the development of conceptual understanding of systems and how they behave. In addition to structuring the work involved in building models, the framework also facilitates easier comparison and evaluation of the results of modelling efforts.

This deliverable is part of the NaturNet-Redime project. The presented methodology is meant to support and structure the activities of the modellers working on case studies in work package 6. To apply the methodology basic knowledge of qualitative reasoning and modelling is required.

Acknowledgement

Tim Nuttle (University of Jena) and Peter Barz (Environmental Network Limited) provided many helpful comments on earlier versions of this deliverable. We also want to thank the participants of the NaturNet-Redime workshop in Amsterdam (October 11-14, 2005) for their feedback and suggestions concerning the methodology.

Document history

Version	Status	Date	Author
01	Draft, initial version.	31/06/2005	Bredeweg .
02	Draft, introduction, section 2.	11/09/2005	Bredeweg.
03	Draft, extended introduction.	13/09/2005	Bredeweg.
04	Draft, running examples, conventions, small details.	20/09/2005	Bredeweg, Bouwer, Liem, Salles.
05	Draft, new and modified text on the general parts of section 3, 4, and 5, correcting typos.	26/09/2005	Bredeweg, Bouwer, Liem, Salles.
06	Draft, typos, added some missing parts, and improvements of the two running examples.	01/10/2005	Bredeweg, Salles Bouwer, Liem.
07	Draft (for internal review), typos, new & modified text for section 6, 7 and 9.	02/10/2005	Bredeweg, Salles.
08	Draft (for Amsterdam workshop), corrected a few typos.	07/10/2005	Bredeweg.
Final	Corrections and typos	25/10/2005	Bredeweg

Contents

1 INTRODUCTION 5

1.1 QUALITATIVE REASONING – A BRIEF INTRODUCTION 5
1.2 A STRUCTURED APPROACH TO MODELLING 6

2 ORIENTATION AND INITIAL SPECIFICATION 9

2.1 DOCUMENTATION 9
2.2 MAIN MODEL GOALS 9
2.3 CONCEPT MAP 10
2.4 RUNNING EXAMPLES 11
2.4.1 Running example: Communicating vessels system 11
2.4.2 Running example: The Ants' Garden 12

3 SYSTEM SELECTION AND STRUCTURAL MODEL 15

3.1 SYSTEM STRUCTURE 15
3.2 SYSTEM ENVIRONMENT AND EXTERNAL INFLUENCES 15
3.3 ASSUMPTIONS CONCERNING STRUCTURE 16
3.4 RUNNING EXAMPLE: COMMUNICATING VESSEL SYSTEM 16
3.5 RUNNING EXAMPLE: THE ANTS' GARDEN 17

4 GLOBAL BEHAVIOUR 19

4.1 PROCESSES 19
4.2 EXTERNAL INFLUENCES AND DELIBERATE ACTIONS 20
4.3 CAUSAL MODEL 20
4.4 SCENARIOS AND BEHAVIOUR GRAPHS 21
4.5 ASSUMPTIONS CONCERNING BEHAVIOUR 22
4.6 RUNNING EXAMPLE: COMMUNICATING VESSEL SYSTEM 22
4.7 RUNNING EXAMPLE: THE ANTS' GARDEN 24

5 DETAILED SYSTEM STRUCTURE AND BEHAVIOUR 27

5.1 STRUCTURAL DETAILS 27
5.2 AGENTS 27
5.3 ASSUMPTIONS 27
5.4 QUANTITIES AND QUANTITY SPACES 28
5.5 DETAILED DESCRIPTION OF SCENARIOS 28
5.6 DETAILED DESCRIPTION OF MODEL FRAGMENTS 28
5.7 RUNNING EXAMPLE: COMMUNICATING VESSEL SYSTEM 29
5.8 RUNNING EXAMPLE: THE ANTS' GARDEN 32

6 IMPLEMENTATION 37

6.1 RUNNING EXAMPLE: COMMUNICATING VESSEL SYSTEM 37
6.1.1 Entities 37
6.1.2 Attributes 37
6.1.3 Agents and assumptions 37
6.1.4 Configurations, quantities and quantity spaces 37
6.1.5 Scenarios 38
6.1.6 Model fragments 39
6.2 RUNNING EXAMPLE: THE ANTS' GARDEN 40
6.2.1 Scenarios 40
6.2.2 Model fragments 41
6.3 MODEL DEBUGGING, SIMULATION AND FINALISING 42
6.3.1 Understanding states 42
6.3.2 State transitions 43
6.3.3 Removing branches 43
6.3.4 Finalisation 44

7 MODEL DOCUMENTATION 45

7.1 CONTEXT, MODEL OBJECTIVES, AND USE OF THE MODEL 45
7.2 GLOBAL STRUCTURE AND BEHAVIOUR 45
7.3 IMPLEMENTATION DETAILS 46

7.4 SIMULATION RESULTS 46
7.4.1 Running example: The Ants' Garden 47
7.5 MODEL EVALUATION 49
7.6 SUMMARY TABLE 50
7.6.1 Running example: Communicating vessel system 51

8 VOCABULARY STANDARDS AND CONVENTIONS 53

9 CONCLUSION 57

10 LITERATURE 59

1 Introduction

Building a model is a process during which potentially vague and general ideas become detailed and formally specified. The goal of this document is to support developers of qualitative models and simulations in performing such a task, particularly to structure the work carried out by the modellers working on case studies in work package 6. This document is part of a set of deliverables that together highlight different aspects of qualitative reasoning and modelling:

- D4.1: Single-user QR model building and simulation workbench (software)
- D4.2.1: User-manual for single-user version of QR workbench (document)
- D6.9: Curriculum for learning about QR modelling (document)
- D6.1: Framework for conceptual QR description of case studies (this document)

D4.1 refers to the software that will be available for capturing and simulating qualitative models. D4.2.1 is the user-manual that explains how to use the software. D6.9 presents a curriculum that modellers can follow in order to learn about essentials of qualitative reasoning and modelling, particularly focussing on the technical details required to actually build qualitative models. D6.1 presents a structured methodology on how to capture qualitative knowledge, particularly focussing the trajectory of developing a detailed model from a general idea. Notice that D6.1 is thus *not* about explaining qualitative reasoning ideas and primitives. In fact, some knowledge of qualitative reasoning and modelling is a prerequisite in order to understand and profit from the ideas discussed in this report.

1.1 Qualitative Reasoning – A brief introduction

Qualitative reasoning is an area of Artificial Intelligence that is concerned with the construction of knowledge models that capture insights domain experts have of the structure of systems and their behaviour (functioning). An important goal is to automate this kind of knowledge (using a reasoning engine) and by doing so to support humans in analysing how the behaviour of a system evolves as time passes. To perform such a task, a qualitative reasoning engine takes a *scenario* as input and produces a *state graph* (or behaviour graph) capturing the qualitatively distinct states a system may manifest (Figure 1).

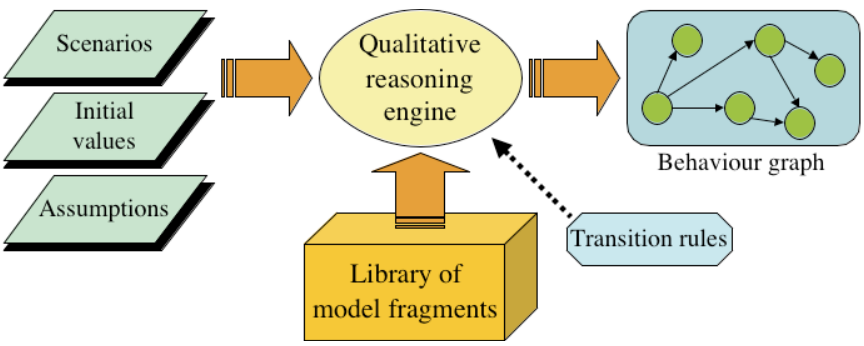

Figure 1: Basic architecture of a qualitative reasoning engine

A scenario usually includes a structural description of the physical appearance of a system, accompanied by statements about initial values and assumptions. A state graph consists of a set of states and state transitions. A state refers to a qualitatively unique behaviour that the system may exhibit (a possible state of behaviour). A state transition specifies how one state may change into another state. A sequence of states, connected by state transitions, is called a behaviour path (a behaviour trajectory of the system). A state graph usually captures a *set* of possible behaviours paths, because multiple state transitions are possible from certain states.

To generate a state graph the engine searches for applicable model fragments from a library. Model fragments can be seen as reusable (conditional) statements that capture knowledge about the phenomena existing in a certain domain. Model fragments applicable to a scenario are assembled by the engine and used to infer the overall behaviour of the system. They are also used to infer the facts that are true in each of the successor states[2]. In general, a model fragment requires certain structural details to be true. If the required structure exists the model fragment is activated for that (partial) structure and introduces the behaviour details that apply to the structure. A specific model fragment can be activated multiple times, namely for each occurrence of the (partial) structure to which it applies. For further details see e.g. Bredeweg et al. (in press).

A fundamental aspect of building a qualitative model is thus the construction of a library of model fragments, for a certain domain (Physics, Ecology, etc.), that can be used to reason about the behaviour of a set of systems belonging to that domain.

1.2 A structured approach to modelling

Building a qualitative model is a complex task. It consists of creating a library of model fragments and accompanying scenarios such that when simulating those scenarios they produce output that answers the questions specified in the modelling goals. This document presents a *structured approach* that defines a protocol to support the execution and management of this modelling task. This structured approach has six steps (see also Figure 2).

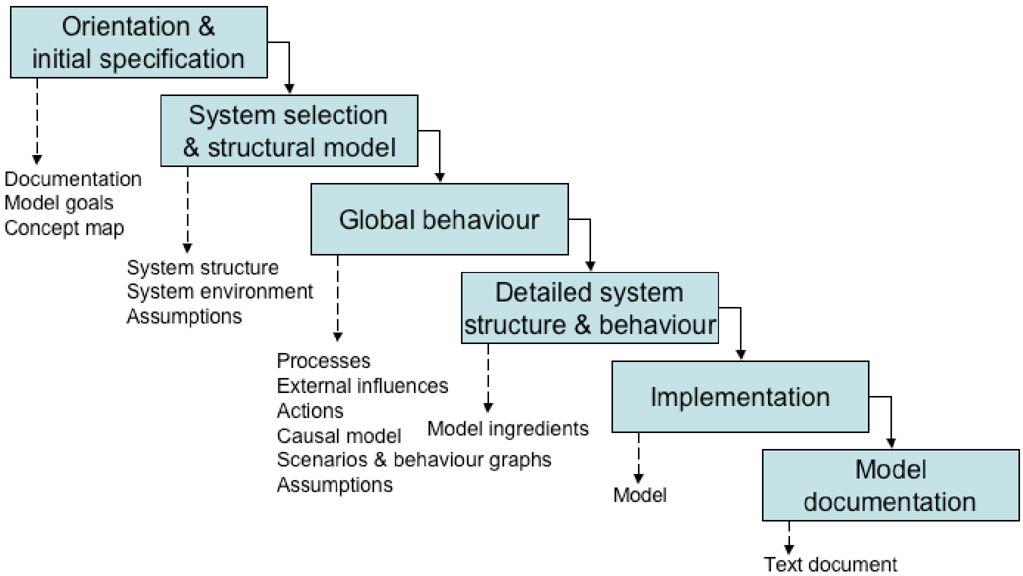

Figure 2: Structured approach to building qualitative models

1. Orientation and initial specification: establishing what should be modelled, why and how.
2. System selection and structural model: identification of the target system structure and its constituents.

[2] This implies, among other things, that the set of facts may change and can be different for alternative states.

3. Global behaviour: general specification of the behaviour that the model should capture.
4. Detailed system structure and behaviour: detailed specification of the behaviour to be captured.
5. Implementation: creation of the model ingredients in the model-building software. Simulation and debugging in order to improve and optimize the model and obtain the required results.
6. Model documentation: documentation of the model and underlying argumentation.

Each of these steps is further detailed in this document, including:
- A general description of the task.
- A formalism to express the content developed during the task.
- Two running examples to further illustrate the basic idea behind each step.

The structured methodology presented in this report is not the only way in which models can be built. However, creating a qualitative model is a difficult task, and experiences in the past have shown that a structured approach, during which the model is step-wise clarified, defined, and documented, creates a momentum that makes success more likely (Salles & Bredeweg, 2003). Moreover, having all the intermediate representations and argumentations explicated significantly reduces the work that has to be done in order to establish a proper documentation of the end-result.

2 Orientation and initial specification

Goal: Developing a broad understanding of the phenomena that will be modelled

Before a model can be built, ideas need to be formulated concerning the contents and purpose of the model. Three aspects are important here. The first aspect is a global textual description of the target system and how it behaves. The second is a concept map that highlights the important concepts relevant to the system being modelled. The third is the goals for which the model is being build, particularly addressing which characteristics of the target system will be captured in the model and how that will be observable in the simulation results.

These three aspects are important references during the model-building process as well as for using the model afterwards because:

- The human mind is a dynamic thing. Having explicit documentation on content and goals will be of help for managing the modelling process.
- When a model is being built as a collaborative activity it is crucial to develop mutual understanding of the target system and its behaviour. This is important before the modelling effort starts, but also during the process to avoid deviation and conflicts among the modellers.
- Model goals provide means to evaluate the model and simulation results during the construction process and when it is finished.
- Model goals provide means for potential users of the model to assess the applicability of the model for their purposes.

2.1 Documentation

Documentation may take on a number of forms, but it generally has two flavours: informative and normative. Informative refers to documents found by the model developers that have been created by others. Such documents help developers to study and understand the target system. Normative documents on the other hand, are created by the model developers themselves, and are the first step to express their intentions about the model they will create. Hence, they are formative for the expected end-result. Typical examples of documentation are:

- Informative documentation
    - Pictures and/or drawings of the target system and its behaviour.
    - Documents describing the system (including URL's referring to websites).
- Normative documentation
    - Summary and overview of the main features written by the model developers.
    - A (short) presentation of the target system (e.g. using PowerPoint), which can be used to explain aspects of the system to 'outsiders'.

2.2 Main model goals

A model is usually created to serve a purpose. The purpose usually includes one of more specific features that the model exhibits and an identification of the target audience (those who will use the model). During the 'orientation and initial specification' step, the formulation of this purpose is by definition general. Also, we do not yet want to restrict ourselves to a particular vocabulary to express these goals. Hence, the goals are expressed in natural language. For a qualitative model typical goals include:

- Developing a knowledge structure that can be used to explain phenomena[3] (e.g. to teach learners, or to support stake-holders in developing an argumentation).
- Determining all possible behaviours of a system.
- Investigating the possibility of phenomena to occur (proof that phenomena can or cannot occur).
- Investigating interactions between phenomena (Are there interactions? What kind of interactions are there? Why are there no interactions?).
- Investigating the co-existence of phenomena (proof that co-existence is possible or impossible).

2.3 Concept map

A concept map (sometimes also referred to as an entity-relation graph) is a graphical representation of ideas that a person believes to be true (Novak & Gowan, 1984), and as such represents knowledge. A concept map consists of two primitives: nodes and arcs. Nodes reflect important concepts, while arcs show the relationships between those concepts. An example of a concept map is shown in Figure 3. Cañas et al. (2003) define concept maps as follows: "Concept maps are graphical representations of knowledge that are comprised of concepts and the relationships between them. We define a concept as a perceived regularity in events or objects, or a record of events or objects, designated by a label".

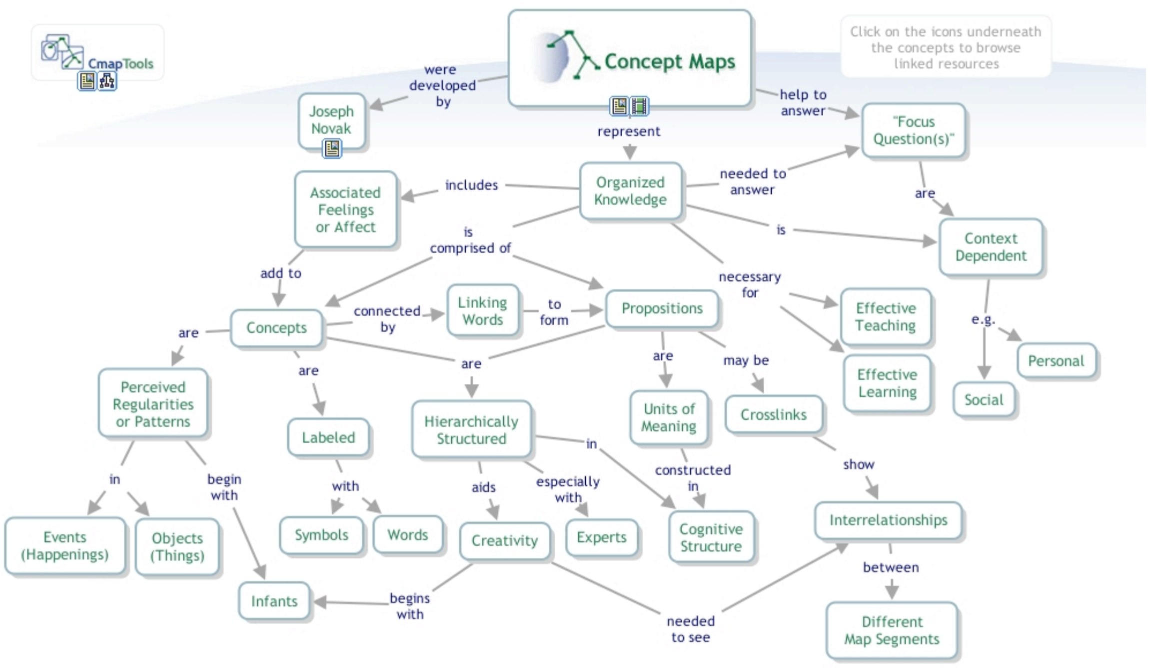

Figure 3: Concept map about concept maps (by Joseph D. Novak)
(http://cmap.ihmc.us/Documentation/ - visited August 31st, 2005)

Making a concept map has at least two purposes:

- Clarifying, organising and developing ideas and insights on behalf of the developer.
- Establishing means to facilitate the development of a shared understanding.

Externally represented concept maps can be used to discuss the content captured with other model builders. During this process the map can be further adjusted to

[3] The word 'phenomena' refers to static and dynamic system behaviours.

accommodate alternative ideas, and may ultimately reflect a shared understanding among model builders. By making concept maps authors not only externalise information, but by doing so they also further specify and organise their own knowledge. Concept maps can also be used to question domain experts about the correctness of the content captured and based on that input modify the map where needed.

2.4 Running examples

To illustrate the model building steps discussed in this report two running examples are used: the Communicating Vessels System (Forbus, 1984) and the Ant's Garden (Bredeweg & Salles, in press).

2.4.1 Running example: Communicating vessels system

Due to the amount of oil used by humans, large storage facilities are required. Oil is stored in large containers, as shown in Figure 4, left. Multiple connected containers are used for this purpose. The connection is made using pipes at the bottom of the containers. Effectively, this transforms the storage facility into a giant system of communicating vessels.

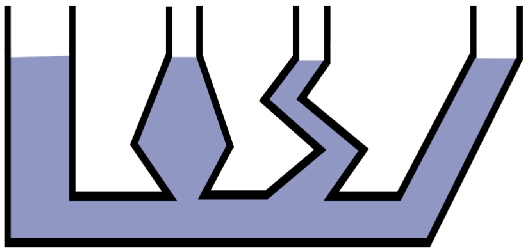

Figure 4: Communicating vessels: Oil containers (left) & Schematised drawing (right)

A schematised version of a communicating vessels system is shown in Figure 4, right. The system consists of a number of vertical containers of different shapes, which are connected at the bottom by a pipe. The fluid has the same height everywhere, provided the containers contain the same liquid. If oil is added to a container, the flow will level the fluid in each of the containers. Increasing or decreasing the amount of oil will affect the height of the oil in all other containers.

This phenomenon occurs, because the fluids, which are acted upon by gravity, cause an equilibrium of the pressures of fluids. These pressure magnitudes depend only on how far their surfaces are from the bottom of each container. This means that the pressure is not affected by the width or the shape of the container (as shown in Figure 4, right), but depends only on the height of the fluid column. In effect, the connected oil containers represent a single vessel, in which the heights of the fluid are equal.

Model goals

The main goal of the communicating vessels model is to teach basic physics concepts to high school students. The system is described in the context of oil containers to make the explanation more interesting for students. The final model has to be kept simple.

Concept map

Using the normative documentation written in the previous section, the relevant concepts and relations in communicating vessel system have been captured in a concept map (Figure 5).

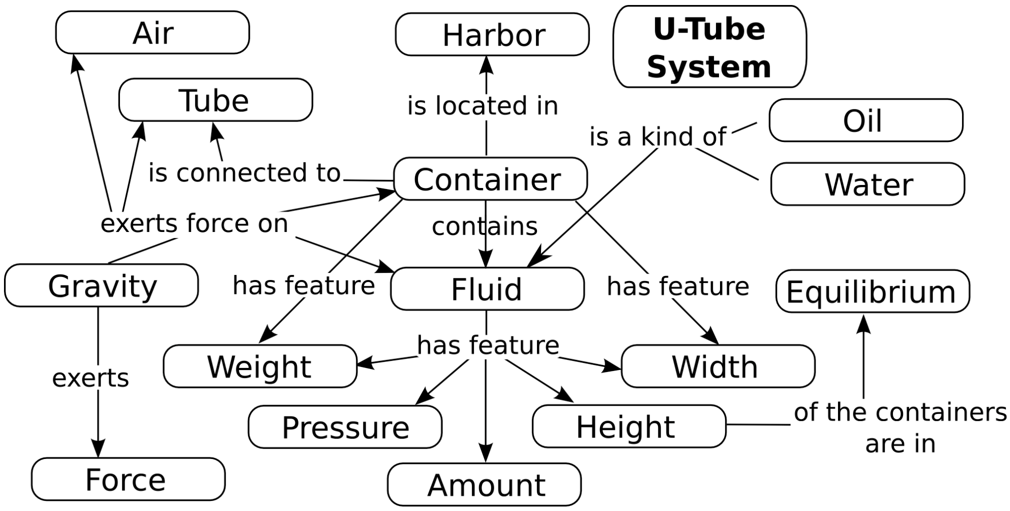

Figure 5: Concept map for the communicating vessel system

2.4.2 Running example: The Ants' Garden

The Ants' Garden is a complex ecological system that has been studied extensively for about a century, and still attracts ongoing research. Ants have been farmers long before people began to plough the earth (see also Figure 6), some estimate since 50 million years ago (Lumpkin & Hsia, 2004). Ants (e.g. Formicidae) grow fungi underground and spend a great deal of effort on keeping the fungi in a healthy state (hence the name "Ants' Garden"). This is an example of a symbiosis from which both species benefit. Trying to disrupt this symbiosis by experimentally poisoning the food (an exogenous influence) shows adaptive behaviour of the ants because they stop feeding the food to the fungi, even when the ants themselves are not able to detect the poison (Fell, 1996).

Figure 6: The leafcutter ant is a type of ant that grows crops of fungus underground. (*Scott Bauer/USDA ARS*), source: http://nationalzoo.si.edu/Publications/ZooGoer/2004/4/antfarmers.cfm

Recent studies (Currie et al., 1999a; 1999b) showed that the Ants' Garden is even more complex and often involves a third species, the specialised garden parasite fungi of the genus Escovopsis which may destroy the system, by attacking the cultivated fungi. However, it almost never happens because ants carry on their body colonies of bacteria (genus Streptomyces) that produce antibiotics effective in controlling the growth of Escovopsis. Therefore the system consists of four species and of complex balance of interactions in which eventually the ants' garden survives. Further studies (Currie et al., 1999a) indicate that the bacteria also produce metabolites (vitamins, amino acids) that may enhance the growth of the cultivated fungi. Considering this, there is yet another interaction, this time between bacteria and cultivated fungi. Finally, sometimes, external actions may influence the system. For example, a researcher may be investigating the effects of pollution on cultivated fungi and its consequences to the whole Ants' Garden system.

Model goals

The goals of building a model for the Ants' Garden are the following:

- By using a qualitative reasoning approach, to provide a formal way for describing the structure and behaviour of the Ants' Garden.
- To build models that are able to support the answering of questions about the mechanisms (at the population level) that keep the Ants' Garden functioning, even in the presence of the parasitic fungi.
- To have a model that can be used to communicate the refined understanding of the Ants' Garden structure and behaviour. When the model has reached a mature level of accuracy, it can be used for scientific purposes in discussions about the Ants' Garden phenomenon.
- To run virtual experiments. Having a runnable qualitative model would enable us to run simulations as a kind of virtual experiments with different versions of the Ants' Garden. For example, varying the number of species and/or the types of behaviour involved would create a number of simulation scenarios. Checking the simulation results with the outcomes of real experiments can teach us about the accuracy of the underlying model and assumptions, and perhaps lead to proposals for new experiments to be performed.
- To build a model that is understandable, manageable and that allows full exploration. That is, a conceptual model that can be used as an educational tool to improve learners' understanding of the Ants' Garden.

Concept map

Figure 7 shows a concept that summarizes the main ideas about the Ants' Garden.

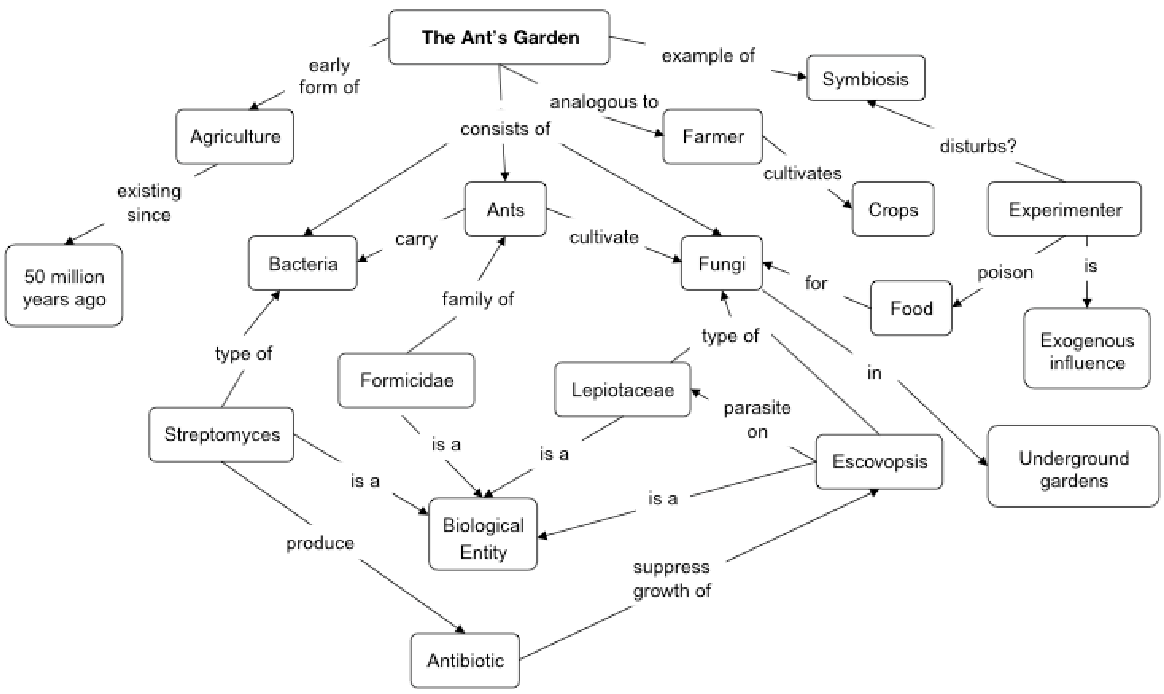

Figure 7: Concept map for the Ants' Garden

3 System selection and structural model

Goal: *Identifying the system structure, particularly the entities involved and how they are related. Distinguishing the system from its environment. Identifying the assumptions made while specifying the structure of the system.*

In the 'orientation and initial specification' step the goal was to broaden the view on the system. The purpose of the 'system selection and structural model' step is to make the initial choices concerning the structure of the system. Which entities will be included in the model and which not? And what are the assumptions made when imposing a certain structural organisation on the system.

3.1 System structure

System structure refers to the physical world as perceived by humans. It refers to those parts of the system that in principle do not change due to the behaviour of the system. When building a model, an important step is thus to determine the structure of the system, particularly to determine the entities from which the system is build. For those things that are part of the system further details need to be specified, namely:

- Type (what kind of entity is it?) – For each entity a description must be made of what it is. Often some entities are of the same type. In that case one description of the general description of the type suffices.
- Structural relationship (how do entities relate to each other?) – For each pair of entities consider whether the entities are structurally related. If they are, provide a description of that relationship.
- Decomposition (does the entity consist of subparts?) – Often objects can be decomposed into a set of other parts (the parts from which the bigger object is made). Identify and describe these decompositions and for the subparts, describe their type, structural relationship, and possible decomposition.

3.2 System environment and external influences

When creating a model certain aspects will not be included, because they are outside of the system boundary. In general, there is a huge collection of characteristics that are not included in a model. The goal is (of course) not to enumerate everything that is not part of the model. However, for certain aspects it is sometimes relevant to mention their exclusion and be explicit about that. These, include aspects that are:

- Irrelevant – When building the model, irrelevant aspects can be ignored. There appears to be no behavioural aspect of the system that is affected by it. They are thus fully irrelevant. Ignoring usually means that they are not included in the model at all. However, it may be worthwhile to describe such aspects in the context of the 'system selection and structural model' step, particularly when the choices about what is considered irrelevant deviate from what is considered normal practice.
- Important under certain conditions – Some details can be ignored until certain boundaries are reached. After that, they need to be taken into account. As a QR engine dynamically builds the model from which it calculates the simulation results, these aspects can be made available when needed. That is, when the conditions become true. As an example consider the possibility of an iron container melting. Only under specific conditions, for instance in the case of exceptionally hot heaters, reasoning about that possibility should be considered. In all other cases it should be ignored.

- External influences – These influences refer to impacts enforced upon a system by aspects that are in principle outside the system as such. Consider a person managing a certain vegetation, e.g. by providing water on a regular basis. In a QR model the vegetation could be considered the system, and the impact of the manager could be presented as an external factor (exogenous) that has a certain impact on the system, but that itself is not a part of the system as such.

3.3 Assumptions concerning structure

Throughout the 'system selection and structural model' step, choices have been made concerning what to include, what to leave out, and how to include it. All these choices must be enumerated, because they help outsiders to assess the model and its simulation results. They are also important for placing a model into perspective with other models.

Some of these assumptions will eventually become model ingredients and play an active role helping the engine to decide what aspects to include in a model and what to exclude. Other assumptions will not be included in the model. They remain textual descriptions of choices made and what impact those choices have on the model and its results.

3.4 Running example: Communicating vessel system

Entities

- **Communicating vessel system**: A communicating vessel system is an object that consists of two containers and a pipe.
- **Oil**: Oil is a fluid that is possibly contained by a container.
- **Water**: Water is a fluid that is possibly contained by a container,
- **Fluid**: A fluid is a kind of material that can be contained by a container.
- **Container**: There are at least two containers, which are objects, in the communicating vessel system, which may contain a fluid.
- **Pipe**: Pipes are devices that connect containers to each other.
- **Air**: The air is a material that pushes constantly on the water column.

External influences

- **Oil ship**: An oil ship is a thing that can fill oil containers.

Assumptions concerning structure

- **The container bottoms are assumed to be at equal heights.** Otherwise, there could be a higher column of water present, and the height of the fluids would become different.
- **The respective lengths of the containers are assumed to be equal.** Which means that there can be no overflow of only one container.
- **The pipe is assumed to be located at the lowest point in the system.** Which means the fluid can flow freely from one container to the other.
- **Each container is assumed to contain the same liquid.** Otherwise, the heights of the containers could be unequal, as the weight of different liquids is different.
- **The containers are assumed to be rigid.** The width of the containers does not change. Changes in the height of the fluid column due to changes in the width of the container do not have to be modelled.
- **The width of the containers is assumed to be equal.** Although width does not really influence the system behaviour.

3.5 Running example: The Ants' Garden

Entities

Entities that play a role in the Ants' Garden are the following:

- **Ants**: we focus on fungus-growing ants, which are predominantly located in the Neotropics. They play a major role in the ecology of the area by fertilising the soil. Ant colonies have a complex organization, centred around a queen, with different types of workers to cut leaves, carry them, and groom the fungi. We will abstract over these differences, however, and treat the populations of ants as a whole.
- **Fungi**: Ants can cultivate different types of fungi, although each colony of farmer ants grows only one type of fungus in their garden. However, in a typical Ants' Garden, also parasitic fungi occur, of the genus Escovopsis. Interestingly, this type of fungus is only known from Ants' Gardens, and not found elsewhere.
- **Bacteria**: The bacteria belong to a group of bacteria from which many antibiotics for human use, such as streptomycin and tetramycin, are derived. The bacteria are concentrated on the underside of ants' bodies, especially of ants that work in the fungus garden.
- **Food**: This is the food that ants feed to the fungi.
- **Metabolites**: These are vitamins and amino acids produced by the bacteria.

Structural relations

- **Symbiosis**: ants and cultivated fungi.
- **Predation**: cultivated fungi and parasitic fungi.
- **Amensalism**: bacteria and parasitic fungi.
- **Commensalism**: ants and bacteria.

External influences

There is only one agent who exerts an exogenous influence on the system:

- **Experimenter**: the person contaminating the food for the fungi.

Assumptions

There are several assumptions relating to our model of the Ants' Garden:

- All ants, cultivated and parasitic fungi, and bacteria populations are closed populations.
- There is no direct interaction between the ants and the parasitic fungi.
- The bacteria damage the parasitic fungi, but not the cultivated fungi.
- There are no external influences to the garden that determine the typical 'garden behaviour'.

4 Global behaviour

Goal: *Specifying the global system behaviour, including: textual descriptions of the main processes, specifying typical scenarios and their expected behaviours, identifying the overall causal model, and identifying the assumptions made while specifying the global behaviour of the system.*

The purpose of the ‘global behaviour’ step is to make the initial choices concerning the behaviour of the system. The restrictions on the format are still limited in this step, allowing modellers to express their ideas as much as possible. However, some focus is required in order to gradually arrive at specific model details. Hence, the global behaviour is specified using the notion of processes, scenarios and behaviour graphs, causal model, and assumptions, which reflect choices and limitations regarding the global behaviour specified.

4.1 Processes

Forbus (1984) defines processes as “… something that acts through time changing the parameters of objects in a situation. Examples of processes include fluid and heat flow, boiling, motion, stretching and compressing.” These examples originate from physics. For other domains, such as ‘sustainable development’, different processes are relevant (e.g., pollution, reproduction, erosion, photosynthesis, etc.).

For the specification of the global behaviour, identification and description of the processes that govern the system behaviour is a key issue. For each process the following aspects should be clarified:

- Name and type – A process has a name and can be classified as belonging to a certain class of processes.
- Collection of entities (partial system structure) – The set of structurally related entities that provide the context in which the process is active. Obviously these entities should be subsets of those entities identified during the ‘system selection and structural model’ step.
- Quantities involved – The dynamic features of the entities (of the previously mentioned collection of entities) that somehow relate to the changes caused by the process. Particularly, those features that play a role in the start and stop conditions, and the effects of the process.
- Start (triggering) conditions – Processes usually do not occur unexpectedly, but happen because some enabling condition is satisfied.
- Effects (what does it change) – The process changes features of entities. Which features are changed and how does this happen?
- Stop (ending) conditions – What are the conditions under which the process becomes inactive again.

As an example consider a heat flow process:

- Name and type – Heat-flow, a kind of ‘flow’ processes. It is for instance comparable to a liquid flow process.
- Collection of entities – Two physical objects structurally related by a heat-path, e.g. a kettle on a stove, the heat-path being the air in between them.
- Quantities involved – Heat and Temperature of the two physical objects and the (heat) Flow facilitated by the heat-path.
- Start conditions – Unequal temperatures. In this case the stove having a higher temperature than the kettle.

- Effects – A Flow of heat causes the heat of the stove to decrease and the heat of the kettle to increase. These changes in heat cause the temperatures of the two objects to change (more heat increases the temperature, and less heat decreases the temperature).
- Stop conditions – As soon as the temperature difference disappears, the heat flow process stops.

4.2 External influences and deliberate actions

Processes are not the sole cause of changes. Consider for instance a hunter who shoots certain animals, or a pump that puts water from one reservoir into another. Changes such as these are usually perceived as 'external influences' affecting a system or deliberate 'actions' carried our by agents.

Although humans perceive these changes as being different from processes, their textual descriptions are largely similar to describing processes. The main difference is the agent or actor. That is, the person or thing that performs the action. Consider for example hunting:

- Name and type – Hunting, relates to deliberate human actions of changing the size and structure of biological populations.
- Agent (actor) – The hunter (a human being).
- Collection of entities – The hunted population (some kind of animals).
- Quantities involved – Size of the population.
- Start conditions – Current size of the population greater than the desired population size.
- Effects – Reduces the size of the population.
- Stop conditions – Current size of the population smaller than the desired population size.

4.3 Causal model

The processes discussed above provide isolated views on the changes happening to the system. When specifying the causal model, the goal is to create an overview of how the effects of processes propagate to other features (quantities) of the system and how processes interact. The representational means used are direct influences (I's) and proportionalities (P's, indirect influences) (Forbus, 1984). Influences refer to flows initiated by processes (e.g. a *flow* of water from a tap increases the amount of water in the bathtub). Proportionalities refer to propagation of these flows to other system features (e.g. an increasing amount of water in the bathtub causes the level of the water to rise, to increase). Influences can be positive (I+) or negative (I–), expressing that a flow increases or decreases the depending quantity (respectively). Proportionalities can also be positive (P+) or negative (P–). A P+ means that the dependent quantity changes in the same direction. A P– means that the dependent quantity changes in the opposite direction.

A causal model thus becomes an interconnected graph (potentially large) in which the nodes represent quantities and the arcs represent direct (I+/I–) and indirect (P+/P–) influences (see e.g. Figure 8 and Figure 9). Notice that such a graph is in principle a kind of concept map as discussed in Section 2.3. It differs from this general notion of a concept map in that the nodes and arcs are now of a specific type (namely, quantities and I/P's, respectively) and thus have specific meaning.

4.4 Scenarios and behaviour graphs[4]

An important step in specifying the global behaviour is the identification of typical scenarios and behaviour graphs. Scenarios refer to initial situations of the system. Such scenarios may trigger processes and include deliberate actions. Hence the system specified in a scenario may change. Behaviour graphs refer to series of continuous changes that the system will go through following such an initial situation. In a way, describing the typical scenarios and behaviour graphs concerns the specification of the expected simulation output. That is, the results the Qualitative Reasoning engine should produce. Notice that the expected simulation output should somehow reflect the 'model goals' mentioned in Section 2.2.

Descriptions of scenarios may include the following aspects:

- Name
- Collection of entities (partial system structure)
- Agents (if any)
- Quantities
- Initial values and (in)equality statements

Descriptions of behaviour graphs typically consist of a chain of changing quantity values and/or (in)equality statements. Consider for instance a scenario in which a pan with water is heated on a stove. A typical behaviour for this situation is shown in Table 1.

Table 1: Example behaviour for water being boiled

State	Values and (in)Equalities	Description
1	$Temperature_{water}$ < $Temperature_{stove}$ $Temperature_{water}$ < $Boilpoint_{water}$	Initially water is below boiling point and the stove is hotter then the water.
2	$Temperature_{water}$ = $Boilpoint_{water}$ $Amout_{water}$ = decreasing $Amount_{steam}$ = increasing	State1 changes into state2 because water reaches boiling temperature and water starts evaporating into steam.
3	$Amount_{water}$ = zero $Temperature_{steam}$ > $Boilpoint_{water}$ $Temperature_{steam}$ = increasing	State2 changes into state3 because all the water has become steam. The steam temperature keeps increasing.

When specifying the global behaviour, the goal is not to produce *the* perfect behaviour graph that includes all the details concerning the changes of the system behaviour. On the other hand, very specific scenarios and detailed behaviour graphs provide better means to evaluate the model and the simulation results it produces. They also help as a source of reference while building the model.

In addition to the sequence of states (reflecting changing system behaviour over time), three aspects are important to specifying behaviour graphs.

- Begin (start) state – These are the initial states at the start of a behaviour graph.
- End state – These are the states at which a change of behaviour stops. One may refer to these states as goal states, in the sense that the goal of the model is to predict the system behaviour and actually arrive at these end states. Hence, it is important to be explicit about what the intended end states of the simulation are.
- Branching – A state of behaviour lasts for a certain amount of time. After that, the behaviour of the system may change into a 'new state of behaviour'. However, often the transition to successor behaviours is not unique and branching may occurs. Branching refers to the situation when more than one state of behaviour

[4] Also referred to as 'Typical situations and behaviours'.

follows a previous state of behaviour. In the example above, state2 follows from state1. Consider the situation in which it is unknown how hot the stove actually is. In that case an alternative branch following state1 could have been a state in which: $Temperature_{water} = Temperature_{stove}$.

A behaviour graph can have multiple begin and end states, and also multiple branches. Furthermore, branches may reunite and even produce cycles where behaviour repeats indefinitely.

4.5 Assumptions concerning behaviour

Throughout the 'global behaviour' step, choices have been made concerning which behavioural aspects to include, how to include them, and what to leave out. All these choices must be detailed, because they help outsiders to assess the model and its simulation results. They are also important for placing a model into perspective with other models.

As with the structural assumptions, some of the behaviour assumptions may eventually become model ingredients and play an active role helping the engine to decide on what aspects to include in a model and what to exclude. Other behavioural assumptions will not be included in the model. They remain textual descriptions of choices made.

4.6 Running example: Communicating vessel system

Processes

- **Liquid flow.** Liquid flow is a typical example of flow process. The process takes place in a system of two containers connected via a tube. The difference in pressure causes a flow from the container with the higher pressure to the container with the lower pressure. The amount of fluid decreases in the former container, and rises in the latter container as a result of the flow. This process starts when there is a pressure difference between the two containers, and ends when the pressures become equal.
- **Pressure of air on liquid.** The water particles are attracted due to gravity, and cause pressure on the fluid surface. This process can be active when there is a container filled with water. This process affects the weight of the air, depending on the amount of air. The weight of the air causes pressure on the liquid in the container. We assume that the effect of the air pressure on the water is negligible. Therefore this process will not be implemented.
- **Pressure of liquid on containers.** The particles in the air are attracted due to gravity, and cause pressure on the containers. This process is active when there is a container filled with water. The pressure of the water exerts a force on the containers. As this process in levelled out by the resistance of the containers, this process in not implemented.
- **Resistance of containers.** The water and the air cause pressure on the container, but the container applies an equal 'reaction force'. Whenever there is a container and a pressure applies a force on a container, the container applies an equal force inward. As the pressure of the water on the containers is not implemented, there is no need to implement this process.

External influences and deliberate actions

- **Oil ship filling the container.** An oil ship could be filling one of the containers in the system. This would increase the amount of fluid in the container. This activity starts whenever the oil ship starts filling the container, and stops whenever either the oil ship stops transferring oil, or the container is full.

Causal model

In the causal model shown in Figure 8, the assumption is made that there is a flow from the container on the left to the one on the right. The reasons for the choice of causal operators are:

- If the amount of water increases, the height increases too, therefore there is a P+ relation between them.
- If the height of water increases, the pressure increases too, therefore there is a P+ relation between them.
- If the pressure on the left side increases, the flow will increase too, therefore there is a P+ relation between them.
- If the pressure on the right side increases, the flow will decrease, therefore there is a P- relation between them.
- If there is a liquid flow, the amount on the left side will decrease, therefore there is an I- relation between them.
- If there is a liquid flow, the amount on the right side will increase, therefore there is an I+ relation between them.
- If the width of the container increases, the height of the water will decrease, therefore there is a P- relation between them.

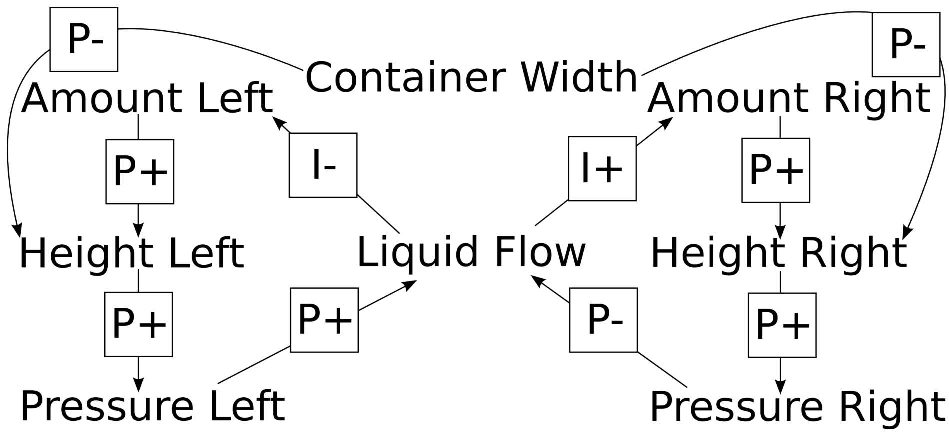

Figure 8: Causal model for communicating vessel system

Typical situations and behaviours

Initial situations that can occur are shown in Table 2.

Table 2: Possible initial situations

Scenario	*Left side*	*Right side*
1.	Full	Full
2.	Full	Some amount
3.	Full	Empty
4.	Some amount	Full
5.	Some amount	Some amount
6.	Some amount	Empty
7.	Empty	Full
8.	Empty	Some amount
9.	Empty	Empty

For these situations the following characteristics can be pointed out:

- In situation 1 and 9 the height of the water in both containers will remain equal.
- In situation 2,3, and 6 there will be a flow from left to right, until both containers have some amount of water, and the heights are equal.
- In situation 4,7, and 8 there will be a flow from right to left, until both containers have some amount of water, and the heights are equal.
- In situation 5 there could be flow either from right to left, or from left to right, or no flow at all, depending on which side the water is higher. There will be a flow from the container with the higher water level to the container with the lower fluid level, until the heights become equal. Or the heights could be already equal, in which case there will be no flow at all.

Assumptions

- **Adhesion and cohesion are assumed to have no effect.** Otherwise, the width of a container might affect the height of the fluid in a container.

4.7 Running example: The Ants' Garden

Processes

The main processes in the behaviour of the populations involved in the Ants' Garden are:

- **Cultivation**: the ants cultivate their fungus by feeding and grooming them. We consider these processes together, as we do not model the difference between the different types of ant workers either. This involves several quantities, such as the size of the ants and fungus populations, and the benefit that they get from it. This process occurs as long as there are populations of ants and cultivated fungi present.
- **Predation**: the parasitic fungi prey on the cultivated fungi. This involves consumption and supply, as well as the number of parasitic fungi being born and the number of parasitic and cultivated fungi dying. This process occurs when there are populations of parasitic as well as cultivated fungi present.
- **Mortality**: populations are negatively affected by mortality, i.e., individuals dying. This process occurs for every existing population.
- **Natality**: populations are positively affected by natality, i.e., new individuals being born. This process occurs for every existing population.
- **Poisoning**: the experimenter poisoning the cultivated fungi. The amount of poison positively affects the number of fungi dying. This process occurs only in experimental conditions, when an experimenter introduces poison to the system.

Causal model

There are two basic processes influencing each population: natality and mortality. In total eight of such processes are active in the Ants' Garden. These processes influence four quantities that represent the amount of each organism, and changes in these amounts propagate to other quantities, used to represent the effects of one population into another population. The decision made with respect to the causal relations can be summarized as follows (see also Figure 9):

- If there is a positive birth rate of ants, the number of ants will increase; therefore there is an I+ relation between them.
- If there is a positive death rate of ants, the number of ants will decrease; therefore there is an I– relation between them.
- If the number of ants increases, the benefit produced by the ants increases too; therefore there is a P+ relation between them.

- If there is a positive birth rate, the amount of cultivated fungi will increase; therefore there is an I+ relation between them.
- If there is a positive death rate, the amount of cultivated fungi will decrease; therefore there is an I– relation between them.
- If the amount of cultivated fungi increases, the benefit produced by cultivated fungi increases too; therefore there is a P+ relation between them.
- If the amount of cultivated fungi increases, the supply produced by cultivated fungi increases too; therefore there is a P+ relation between them.

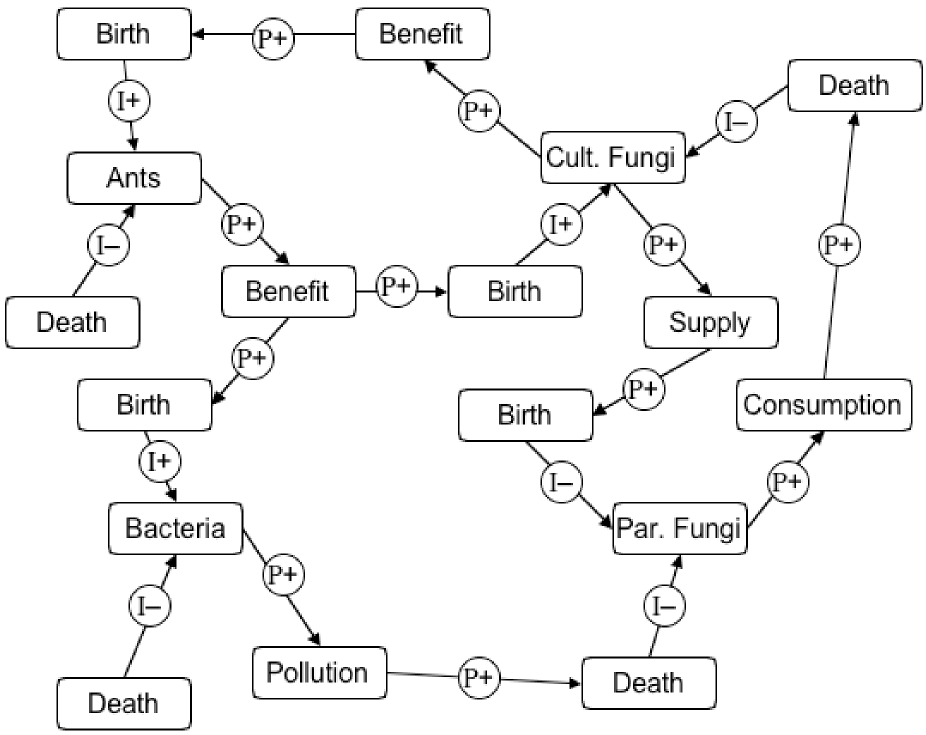

Figure. 9. Causal model for The Ants' Garden

- If there is a positive birth rate, the amount of bacteria will decrease; therefore there is an I– relation between them.
- If there is a positive death rate, the amount of bacteria will decrease; therefore there is an I– relation between them.
- If there the amount of bacteria increases, the pollution produced by the bacteria increases too; therefore there is a P+ relation between them.

- If there is a positive birth rate, the amount of parasitic fungi will increase; therefore there is an I+ relation between them.
- If there is a positive death rate, the amount of parasitic fungi will decrease; therefore there is an I– relation between them.
- If the amount of parasitic fungi increases, the consumption due to parasitic fungi increases too; therefore there is a P+ relation between them.

- If the benefit produced by the ants increases, the birth rate of cultivated fungi increases too; therefore there is a P+ relation between them.
- If the benefit produced by the ants increases, the birth rate of bacteria increases too; therefore there is a P+ relation between them.

- If the benefit produced by cultivated fungi increases, the birth rate of ants increases too; therefore there is a P+ relation between them.
- If the supply due to cultivated fungi increases, the birth rate of parasitic fungi increases too; therefore there is a P+ relation between them.

- If the consumption due to parasitic fungi increases, the death rate of cultivated fungi increases too; therefore there is a P+ relation between them.

- If the pollution due to bacteria increases, the death rate of parasitic fungi increases too; therefore there is a P+ relation between them.

Typical situations and behaviours

When observing Ants' Gardens in nature, they may exhibit several different behaviours. We wish to model several typical situations and associated behaviours:

- The four populations live together: the four populations may reach stability with different combinations of values, and continue in a stable manner, without changes.
- Parasitic fungi disappear: Bacteria may eventually destroy the parasitic fungi population, and then the three populations (ants, bacteria, cultivated fungi) continue living together.
- Bacteria disappear: Bacteria would disappear and leave the other three populations co-existing and stable within a range of normal sizes (this situation is not found in nature, yet).
- Parasitic fungi and Bacteria disappear: since the relation between the ants and the bacteria is adaptive when the parasitic fungi are present, it may happen that cultivated fungi and ants coexist without the parasitic fungi and bacteria.
- The whole garden disappears: extinction of the whole garden may happen in a situation in which all the populations decrease and are not able to reverse this tendency.

5 Detailed system structure and behaviour

Goal: *Specifying and detailing the system structure and behaviour using Qualitative Reasoning (Garp3) vocabulary.*

The 'detailed system structure and behaviour' step concerns the specification of *all* the model ingredients of the model using the Qualitative Reasoning vocabulary[5].

In principle QR modelling software could be used to do this specification. However, by not yet using the software model builders have more freedom. This freedom is considered essential at this stage of the model building process, because many details still need conceptual detailing and adding implementation constraints may hamper this process. Advantages of ignoring implementation details at this point are:

- The order in which the ingredients are defined is unconstrained
- Small inconsistencies present no direct problems
- Modifying ingredients is relatively easy

When specifying the model ingredients it is worthwhile to not only enumerate them, but to also explain the purpose and meaning of each ingredient in the model.

5.1 Structural details

For the structural details the following ingredients need to be specified:

- Entity types
- Entity supertype relations
- Attributes and their sets of attribute values
- Configurations

Type refers to the kind of entities that may exist. Supertype relationships are used to organise the entity types in a type hierarchy. Attributes and value sets refer to static (non-changing) features of entities. Configurations are binary relationships that specify structural relationships between entities.

5.2 Agents

External influences on a system are represented by:

- Agent types
- Agent supertype relations

Type refers to the kind of agents that may exist. Supertype relationships are used to organise the agent types in a type hierarchy.

5.3 Assumptions

Alternative viewpoints on structural or behavioural aspect can be represented by:

- Assumption types
- Assumption supertype relations

Type refers to the kind of assumptions that may exist. Supertype relationships are used to organise the assumption types in a type hierarchy.

5 For a description of the Qualitative Reasoning vocabulary see for instance D6.9 (Curriculum for learning about QR modelling).

5.4 Quantities and quantity spaces

Changeable features are specified by means of:

- Quantities
- Quantity spaces

Quantities represent the changeable features of entities and agents. How these features can change is determined by their quantity spaces, which represent the possible values a quantity may take on. Quantity spaces are ordered sets of alternating points and intervals. Critical entries in quantity spaces are sometimes called landmarks.

5.5 Detailed description of scenarios

Scenarios describe initial situations. The following ingredients can be specified:

- Name
- Assumptions
- Agents
- Instances of Entities
- Attributes (and specific values)
- Configurations
- Quantities
- Initial values[6]
- (In)equality statements

Name is a unique identifier. Assumptions are labels that may facilitate the inclusion or exclusion of model fragments. Agents represent external influences. Instances of entities, possibly with attribute/value pairs and configurations, represent the structural details. Quantities represent changeable features of entities and agents, and can be given initial values. (In)equality statements can be used to further differentiate between quantities and values.

5.6 Detailed description of model fragments

Model fragments describe chunks of knowledge that may apply to scenarios. For a model fragment the following ingredients can be specified:

- Name
- Supertype
- Conditions
  -
  - Assumptions
  - Agents (only in model fragments of type 'agent')
  - Entities
  - Attributes (and specific values)
  - Configurations
  - Quantities
  - Values
  - (In)equality statements
  - Model fragments
- Consequences
  - Quantities
  - Values

[6] Value refers to both the magnitude and derivative (the direction of change) of a quantity.

   - (In)equality statements
   - Correspondences
   - Influences
     - Direct (I-/I+, only in model fragments of type ‘process’ or ‘agent’)
     - Indirect (P+/P-, also referred to proportionalities)

Conditions refer to ingredients to which a model fragment applies. When the conditions are true the consequences provide additional details that apply to the situation. Name is a unique identifier. A model fragment can be a subtype of ‘static’, ‘process’, ‘agent’, or of an already defined model fragment. Assumptions represent viewpoints and further refine the applicability of a model fragment. Agents represent external influences. Instances of entities, possibly with attribute/value pairs and configurations, represent the structural details. Quantities represent changeable features of entities and agents, and can be assigned values. (In)equality statements can be used to further differentiate between quantities and values. Model fragments refer to chunks of knowledge. Correspondences refer to co-existing values (or sets of values). Influences specify changes and propagation of changes.

5.7 Running example: communicating vessel system

Structural model

The entity subtype hierarchy for the communicating vessels system consists of the following entities (see also Figure 10):

- Entity
  - Object
    - Communicating Vessel System
    - Tube
    - Container
  - Material
    - Fluid
      - Water
      - Oil

The configuration in the communicating vessels system:
- Connected to
- Contains

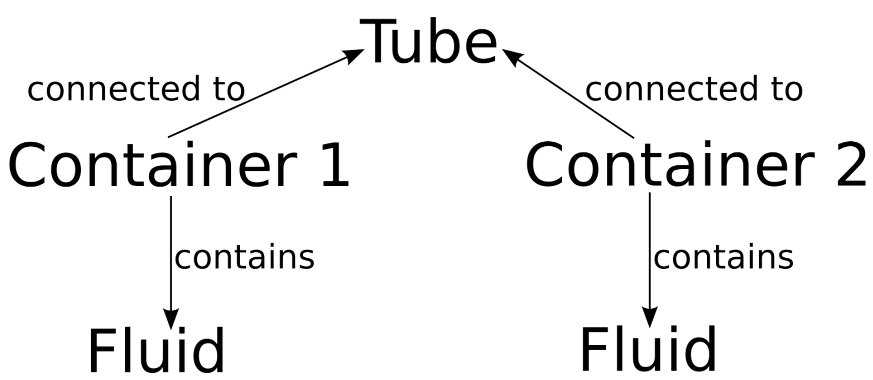

Figure 10: Structural model of the communicating vessels system.

Agents
- The only agent in the system is the oil ship, which is a subtype of *agent*.

Assumptions

- It is assumed that the containers have equal heights and that overflow does not happen.

Quantities, Quantity spaces and landmarks

- **Height.** Height is the quantity that is visible when looking at a container, and is therefore the focus of the simulation. The height quantity specifies the height of the water in a container. From a qualitative point of view there are three interesting values the height quantity can have: empty, positive, or full. Both empty and full are points, positive is an interval in between.
- **Amount.** The amount specifies how much water is in a container. Given the choice of the height quantity space it seems logical to chose an equivalent one for amount. If the magnitude of the amount quantity is full, the magnitude of the height quantity must be full too, empty should result in zero, and positive in positive.
- **Pressure.** The pressure specifies the hydrostatic pressure of the water column in a container. Given that the pressure is caused by the height of the water, it would be a logical choice to choose the same quantity space for pressure. So either no pressure, some pressure, or maximum pressure.
- **Weight.** The weight specifies the weight of the water in a container. As the influence of the weight is captured by the hydrostatic pressure, this quantity can be ignored.
- **Flow.** The flow describes the water flowing from one container to the other through the tube. The water can flow from right to left or from left to right or not flow at all. The value *pos* could indicate flow from right to left, *zero* no flow, and *neg* could indicate a flow from right to left.
- **Width.** The width of the container is relevant as it influences the height of the water in the container. It seems logical the width could be either minimal, maximal, or a value in between.
- **Gravity.** The gravity affects the weight of the fluid depending on the amount of fluid in the container. As weight is captured by pressure, the gravity quantity can be removed from the model.

A summary of the quantities and their quantity spaces is shown in Table 3.

Table 3: Quantity spaces for the communicating vessels

Quantity	***Quantity Space***
Height	{empty, positive, full}
Amount	{empty, positive, full}
Pressure	{empty, positive, full}
Flow	{neg, zero, pos}

Detailed description of scenarios

Each scenario contains two containers (*container1* and *container2*), which are both related to a *tube* using a *connected to* relation. Each container has a *contains* relation to an instance of water (*water1* and *water2).* In the scenario only the height quantity is specified, which is fixed to a specific value.

Figure 11 shows all possible behaviours of the communicating vessel systems. Each state in the graph corresponds to a possible scenario, and following the arrows results in the behaviour for that scenario. The correspondences between the states in Figure 11, and the scenarios in Table 3 are shown in Table 4

Table 4

State	1	2	3	4	5	6	7	8	9	10	11
Scenario	3	2	6	7	4	8	5	5	5	9	1

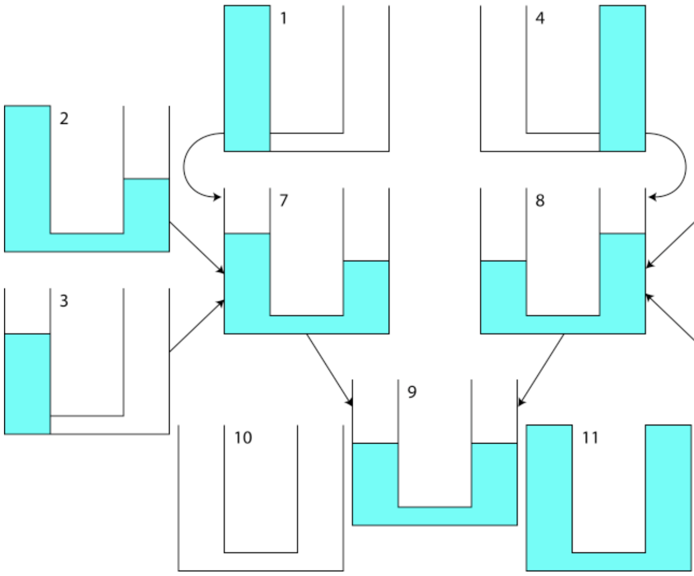

Figure 11: All possible system behaviours

Detailed description of model fragments

Static Fragments

- **Container with fluid:** This model fragment describes just one *container* with a *contains* relation with *water*. Together with the *height* quantity these elements constitute the conditions of the model fragment. As consequences the quantities *amount* and *pressure* are added. There proportionalities described in the causal model are formalised in this model fragment. It is a good idea to make the quantity spaces correspond to each other, to force that *amount*, *height*, and *pressure* change simultaneously (i.e., cannot have different values during simulation).

Process Fragments

- **Liquid flow:** In this model fragment two *Container with fluid* model fragments are imported as *conditions*. A pipe is added as another condition and connects the two containers via the *connected to* structural relation. The pipe has a quantity *Flow* that is added as a *consequence*. Now the rest of the proportionalities and the influences from the causal model can be formalised in this model fragment as consequences. The magnitude of the flow can be derived by subtracting the pressure on the left side from the pressure on the right side. The flow influences the amount of liquid in the left container negatively and the amount in the container on the right positively.

Assumptions

- **No overflow.** The heights of the containers are assumed to be equal. Which means that there can be no overflow.

5.8 Running example: The Ants' Garden

For the description of the behaviour in detail, we choose to focus on the system without the experimenter agent poisoning the food.

Structural model

The following should be modelled as entities and structural relationships.

Entities:

- Ants: organisms from the family *Formicidae*
- Cultivated Fungi: organisms from the family *Lepiotaceae* – the cultivated fungi
- Parasitic Fungi: organisms from the genus *Escovopsis* – the specialised garden parasite fungi
- Bacteria: organisms from the genus *Streptomyces*

Configurations:

- Symbiosis: a concept that represents an interaction between two populations in which both receive benefits from the other population
- Predation: a concept that represents an interaction between two populations in which one population (the prey) is harmed by the second population (the predator), and the latter receives benefit from the former
- Commensalism: a concept that represents an interaction between two populations in which one receives benefit from the second population and the latter is not influenced by the former
- Amensalism: a concept that represents an interaction between two populations in which one is harmed by the second population and the latter is not influenced by the former

Assumptions

- Closed population: an assumption that ignores immigration and emigration in the population, that is, it changes only due to natality and mortality

Quantities, Quantity spaces and landmarks

We need the following quantities for each of the species:

- Born: the amount of new individuals being born
- Dead: the amount of individuals dying for whatever reason
- Number_of: the amount of individuals in the species' population

In addition, the following quantities are required to specify the interactions:

- Benefit: the positive impact produced by the one population on the second population; for example, the benefit produced by the ants on the cultivated fungi population; ants on bacteria; cultivated fungi on ants
- Supply: the amount of resources (food) provided by the prey that may be used by the predator population
- Consumption: the amount of prey resources (food) used by the predator
- Pollution: the degree of pollution by bacteria (or a human experimenter)

The quantities should have the quantity spaces attached to them as shown in Table 5. As we assumed that a population cannot grow beyond a certain maximum, we need a value 'max' in its quantity space for Number_of. It can become zero when the population dies out. As we do not know any other landmarks of interest, we add only the interval in between, which we call the 'normal' range. So, together this constitutes a quantity space QS={zero, normal, max}. For the interaction effect sizes, we use the same quantity space QS={zero, normal, max}, because the distinction between a normal range and a maximum value seems relevant here too. For the quantities Born and Dead, we use a smaller quantity space QS={zero, plus}, because these are not the quantities of prime interest, so not much detail is needed here.

Table 5: Quantity spaces for the Ants' Garden system

Quantity	**Quantity Space**
Number_of	{zero, normal, max}
Benefit	{zero, normal, max}
Supply	{zero, normal, max}
Consumption	{zero, normal, max}
Pollution	{zero, normal, max}
Birth rate	{zero, plus}
Death rate	{zero, plus}

Detailed description of scenarios

First, we consider a number of simple scenarios, with only two populations interacting. This way, the different kinds of interaction (mutualism, Parasitism, amensalism, commensalisms) are exemplified.

Pair-wise interactions:

- Ants & Cultivated fungi
- Cultivated fungi & Parasitic fungi
- Parasitic fungi & Bacteria
- Bacteria & Ants

For all these combinations of entity pairs, the appropriate configurations apply, as mentioned in the structural model. It is possible to create scenarios with different initial values, e.g. each population size (quantity *number_of*) can start off at a value of zero, normal, or max. The most common cases are scenarios with both population sizes having an initial value of normal. We assume that all populations are closed populations, i.e., there is no immigration or emigration.

Complex scenarios involve more species and interactions between them:

- Four interactions between species:
  Ants, Cultivated fungi, Parasitic fungi, and Bacteria
- Five interactions between species:
  Ants, Cultivated fungi, Parasitic fungi, Bacteria, and Metabolites

For these more complex combinations of entities, all the appropriate configurations apply, as mentioned in the Structure model. In the case of four interactions, two scenarios are of special interest: (1) all population sizes having an initial value of normal, and (2) all population sizes having an initial value of normal, except the parasitic fungi (initial value of zero). In the case of five interactions, we consider only the scenario with all initial values being normal. We assume that all populations are closed populations, i.e., there is no immigration or emigration, In addition, we assume that commensalism has a high impact.

Detailed description of model fragments

The model fragments for the ants' garden are classified as static model fragments, basic process model fragments, and interaction process model fragments. Because the ants' garden model is quite complex in nature, it involves many model fragments. A few of the most typical model fragments are described in detail here, only mentioning others.

Static model fragments

- Existing population: Population
  A population exists when its population size is greater than zero
  CONDITIONS:
  Population: Number_of > 0

This model fragment is useful because its presence indicates whether a population exists or not.

Basic process model fragments

There are model fragments for the basic processes such as natality and mortality. Below, the contents of the natality process are presented.

- Natality
  The population increases due to new individuals being born
  CONDITIONS:
  Existing population: Population
  CONSEQUENCES:
  Add quantity: Born
  Add dependencies:
  Born > zero
  Born has a positive influence (I+) on Number_of

The mortality process is defined analogously in another model fragment (not shown), with quantity Dead instead of Born, and a negative influence instead of a positive one.

Interaction process model fragments

The model fragments that model interactions between species are slightly more complex. Below are presented the model fragments for predation and symbiosis, as examples; there are other model fragments for Predation/Parasitism, Commensalism, and Amensalism (not shown), which are defined in a similar manner.

- Predation/Parasitism
  One species benefits by consuming the other species (Supply), which decreases
  CONDITIONS:
  Existing population: Predator
  Existing population: Prey
  Predation: ParasiticFungi
  Prey: CultivatedFungi

  CONSEQUENCES:
  Add quantities: Supply and Consumption
  Add dependencies:
  Prey: Number_of propagates positively (P+) to Supply
  Supply propagates positively (P+) to Predator: Born
  Supply propagates negatively (P-) to Predator: Dead
  Predator: Number_of propagates positively (P+) to Consumption
  Consumption propagates positively (P+) to Prey: Dead
  Supply >= Consumption

- Symbiosis
  Both species benefit from each other (sometimes, symbiosis is viewed as encompassing multiple forms of interaction; *mutualism* is the specific form intended here).
  CONDITIONS:
  Existing population: SymbiontA
  Existing population: SymbiontB
  SymbiontA: Ants
  SymbiontB: CultivatedFungi
  CONSEQUENCES:
  Add quantities: BenefitA and BenefitB
  Add dependencies:
  SymbiontA: Number_of propagates positively (P+) to BenefitA
  BenefitA propagates positively (P+) to SymbiontA: Born
  BenefitA propagates negatively (P-) to SymbiontA: Dead
  And, the same for SymbiontB:
  SymbiontB: Number_of propagates positively (P+) to BenefitB
  BenefitB propagates positively (P+) to SymbiontB: Born
  BenefitB propagates negatively (P-) to SymbiontB: Dead
  Plus:
  ∂BenefitA) corresponds to ∂BenefitB

Assumptions
For the processes comensalism, amensalism, predation and symbiosis, the following dependencies are assumed:
- ∂Number_of determines ∂Born for both populations involved
- ∂Number_of determines ∂Dead for both populations involved

Furthermore, we have the closed population assumption for all populations involved. There are no effects from outside the population (e.g., immigration or emigration), so Birth and Mortality processes are the only important determinants for the population size. This is modelled by the following dependencies:
- Population: Number_of propagates positively (P+) to Population: Born
- Population: Number_of propagates positively (P+) to Population: Dead
- ValueCorrespondence: Number_of = zero corresponds to Born = zero
- ValueCorrespondence: Number_of = zero corresponds to Dead = zero

6 Implementation

Goal: Implementation of the model, that is specifying all the model ingredients in the QR workbench (Garp3) such that each of the scenarios produces the desired behaviour when simulating the model.

Now that in principle all the model ingredients have been specified, the next step is to actually create the model using the Garp3 software. For details on how to use Garp3, see D4.2.1 (User-manual for single-user version of QR workbench). By creating the model errors and inconsistencies may become apparent and will have to be fixed. In addition, running the model will generate results that are not always as expected, and modifications may be needed to improve the model.

Below screenshots are shown to illustrate the possible result of the implementation step for the two running examples. Section 6.3 discusses initial ideas on how to proceed with model simulation and debugging.

6.1 Running example: Communicating vessel system

6.1.1 Entities

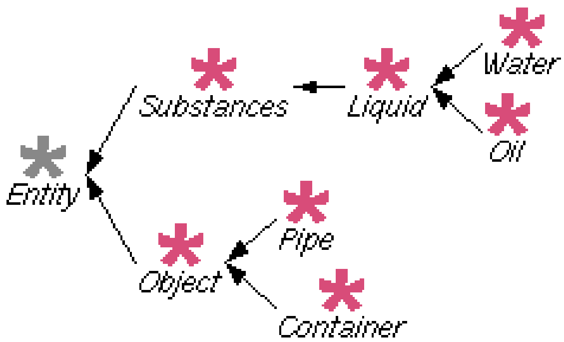

Figure 12: Communicating vessels: Entity hierarchy

The entity hierarchy depicted in Figure 12 shows two classes of entities: substance and object. There are two objects: pipe and container. For substance, two kinds of liquids are represented: water and oil. This means that water and oil inherent any features represented for their super-types: substances and liquid.

6.1.2 Attributes

This model does not include any explicit attributes.

6.1.3 Agents and assumptions

This model does not include any explicit agent and assumptions.

6.1.4 Configurations, quantities and quantity spaces

Figure 13 shows screenshots of the configurations, quantity and quantity space editors. Four quantities have been defined: amount, flow, height, and pressure. These quantities use the quantity spaces QS = {Zero, Plus, Max} and QS = {Min, Zero, Plus}. Finally, four structural relations are defined: connected, contains, from, and to.

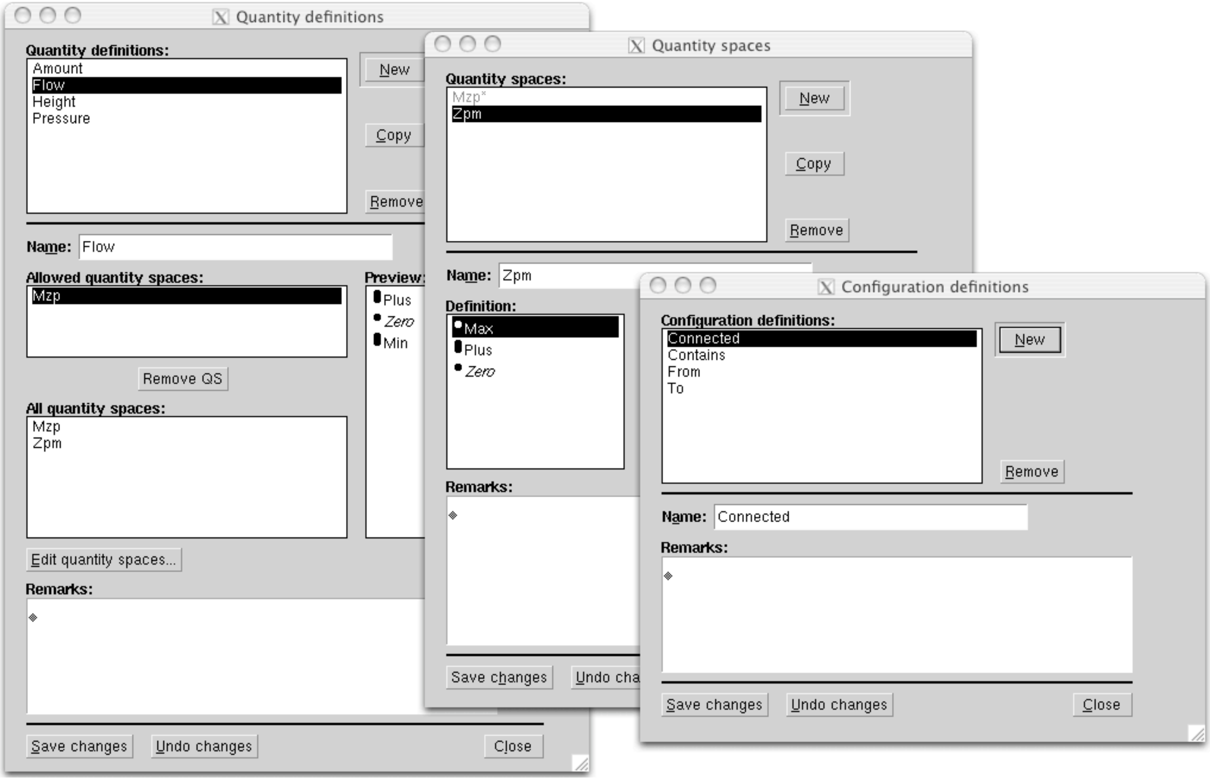

Figure 13: Communicating vessels: Quantities, Quantity spaces, and Configurations

6.1.5 Scenarios

The model has one scenario, which is shown in Figure 14. It specifies two containers, named ‘container left’ and ‘container right’, who each contain oil, ‘oil left’ and ‘oil right’ respectively. The containers are connected to the pipe via the structural relations ‘from’ and ‘to’. Notice that this implies a direction. A flow of oil from the left container to the container on the right will be referred to as a ‘positive’ flow, while a flow in the other direction will be called a ‘negative’ flow. The scenario specifies two initial quantities, namely the height of both columns of oil. These heights have been given the value ‘plus’, meaning that in both containers the oil column has a certain magnitude.

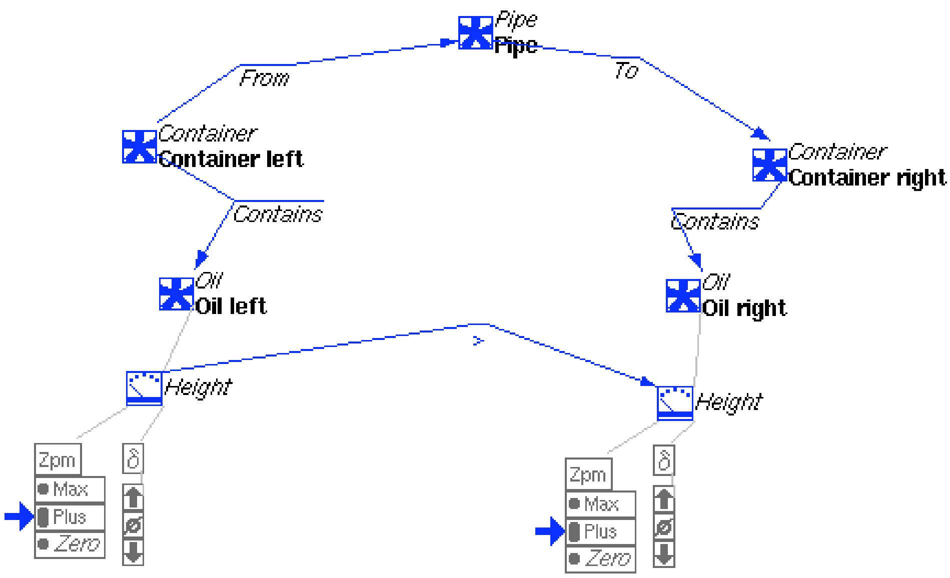

Figure 14: Communicating vessels: A possible scenario

6.1.6 Model fragments

The model has two model fragments. The static model fragment ‘contained liquid’ (Figure 15) and the process fragment ‘liquid flow’ (Figure 16).

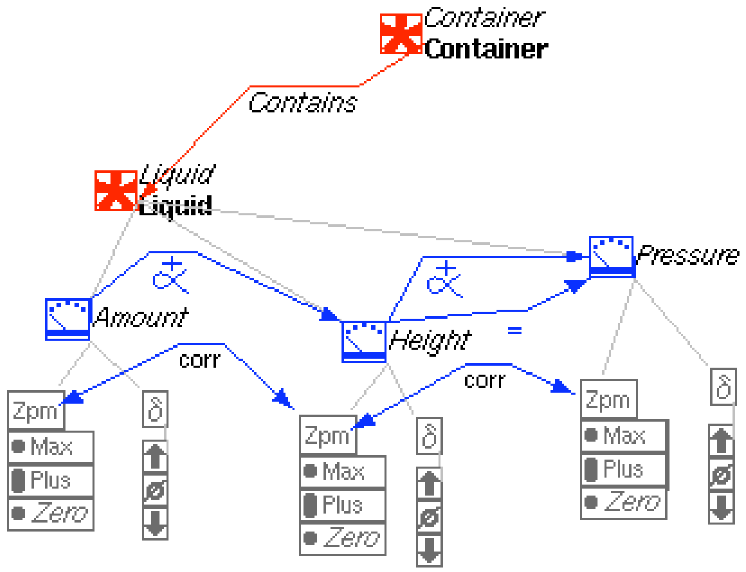

Figure 15: Communicating vessels: Contained liquid

The contained liquid applies to structural details consisting of an entity of the type container that contains an entity of the type liquid (coloured red). When such a structure is found, three quantities are introduced: amount, height, and pressure (coloured blue). The quantities have proportionalities, such that changes in the amount propagate to the height, which in turn propagate to the pressure. Similarly the quantities have correspondences, meaning that the magnitudes for the three quantities correspond. Finally, height is ‘qualitatively equal’ to pressure. By using this, information on differences between comparable heights may propagate to differences between comparable pressures. No specific values (neither magnitudes nor derivatives) are specified.

Figure 16 shows the essentials of the liquid flow process.

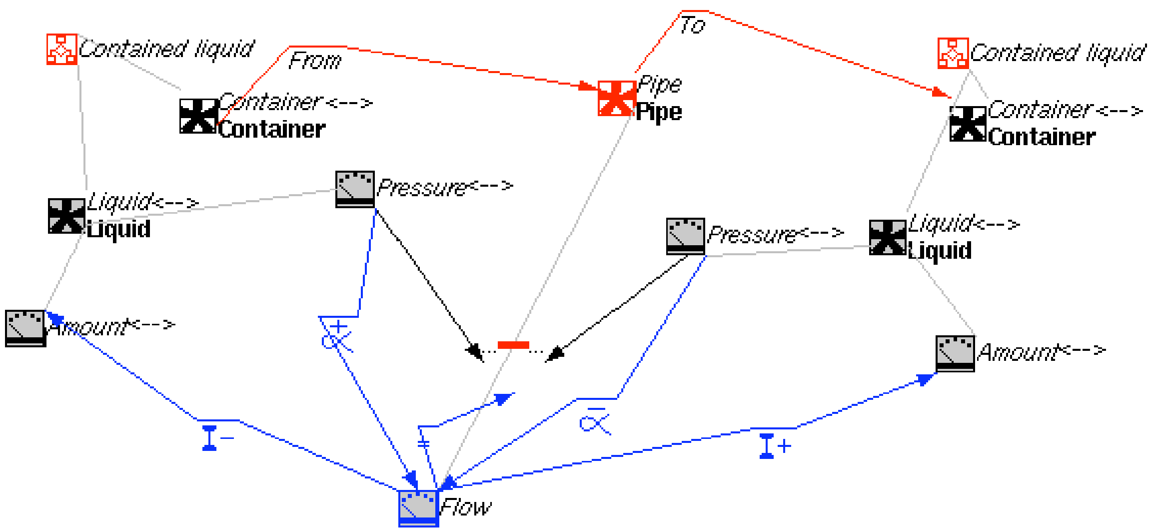

Figure 16: Communicating vessels: Liquid flow process

When this structure exists the quantity flow is introduced which is calculated by subtracting the two pressures (left minus right). The flow negatively influences the amount in the left container and positively the amount in the right container. There is 'feedback' from the pressures on the flow. When the difference changes, the flow should also change. Hence, the pressure 'on the left' has a positive proportionality with the flow and the pressure 'on the right' a negative one. No values are specified.

6.2 Running example: The Ants' Garden

The full model of Ants' Garden system is relatively large. It contains for instance more then 50 model fragments. Enumerating all these details is beyond the scope and purpose of this report. For details see for instance Bredeweg & Salles (in press) and Salles et al. (in press). Below only a small selection of those details are shown.

6.2.1 Scenarios

Figure 17 depicts the details for the most typical scenario of the Ants' Garden. It reads as follows. There are four populations[7]: ants, cultivated fungi, bacteria, and parasitic fungi. Each of the populations starts with *Number_of = <normal, ?>.* There are four interactions: symbiosis, commensalism, amensalism, and parasitism. For commensalism, ants are the producers and bacteria are the affected. For amensalism, bacteria are the producers and parasitic fungi are the affected. For parasitism, the cultivated fungi are the host and the parasitic fungi the parasite. Finally, symbiosis has as symbiont one the ants and as symbiont two the cultivated fungi.

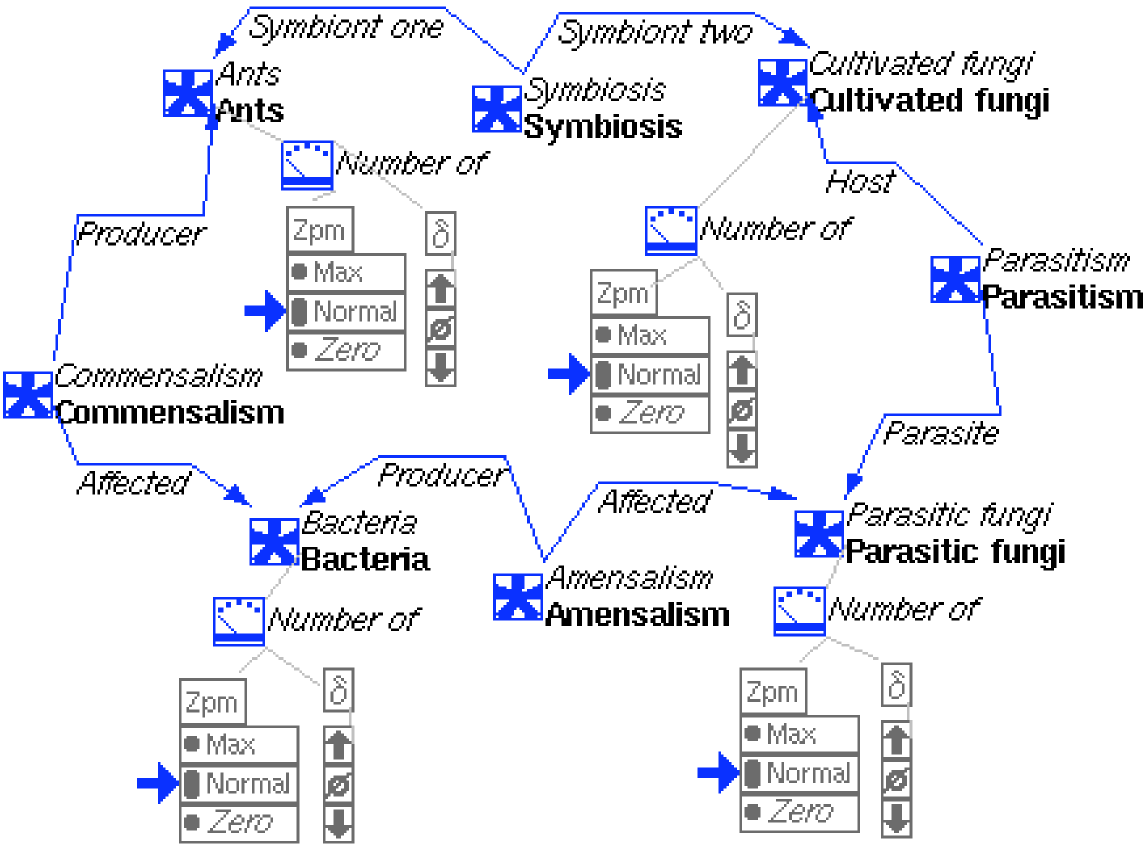

Figure 17: Ants Garden: A typical scenario

[7] The fact that the ants and the other organisms are populations can be found in the subtype hierarchy. This hierarchy is not shown in the Figure.

6.2.2 Model fragments

Figure 18 shows the natality process. It specifies that in the case of an existing population, the birth-rate (born) is positive has a positive influence on the size of the population (number_of).

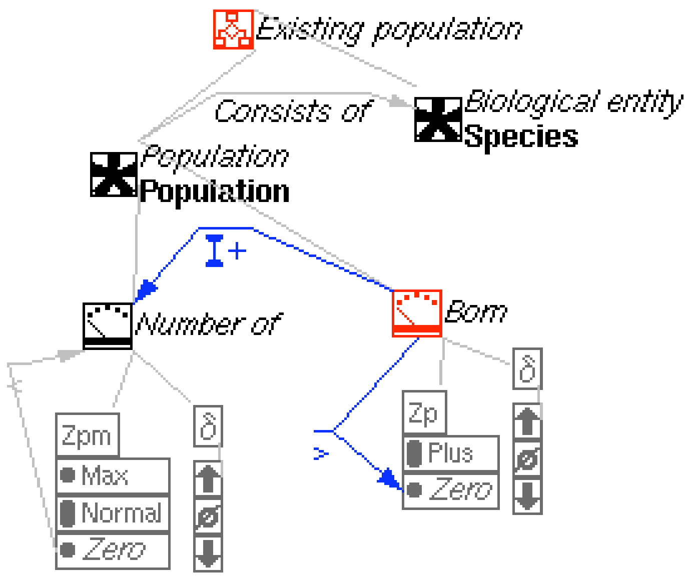

Figure 18: Ants Garden: Natality process

Figure 19 shows the model fragment that recognises a predation situation. If that situation exists, the quantities Consumption and Supply are introduced. Both quantities are proportional to the population size to which they belong. In the model of the Ants' Garden parasitism is modelled analogously to predation.

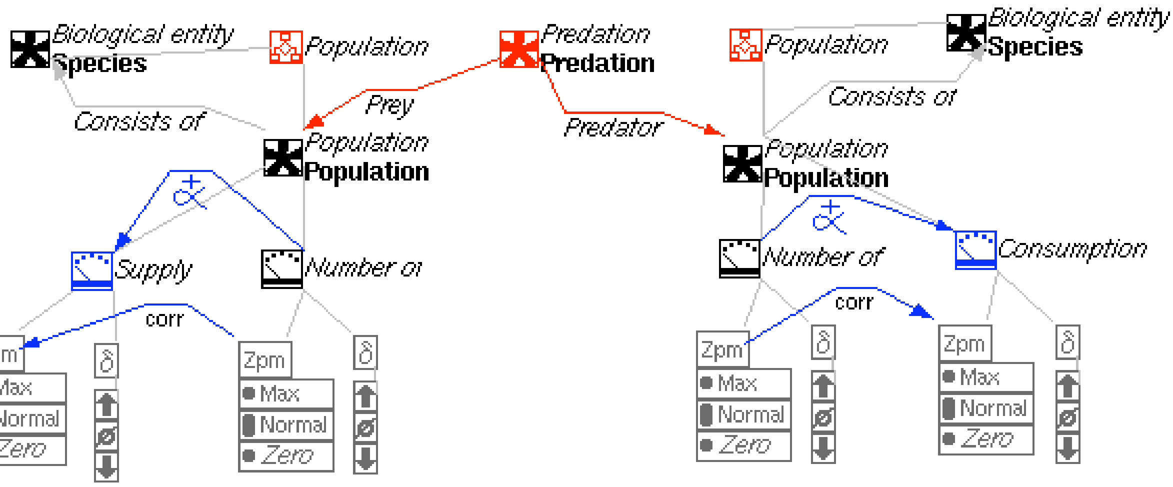

Figure 19: Ants Garden: Predation / Parasitism quantities

Figure 20 shows the interactions that apply when predation (or parasitism) is active. The quantity consumption has a proportionality relation with the death rate of the prey population (the cultivated fungi). The quantity supply has a proportionality relation with the birth and death rate of the predator population (parasitic fungi). Finally, the model fragment specifies details concerning qualitative equality between the magnitudes of the populations involved. For instance, the maximum value for consumption is qualitatively equal to the maximum value for supply.

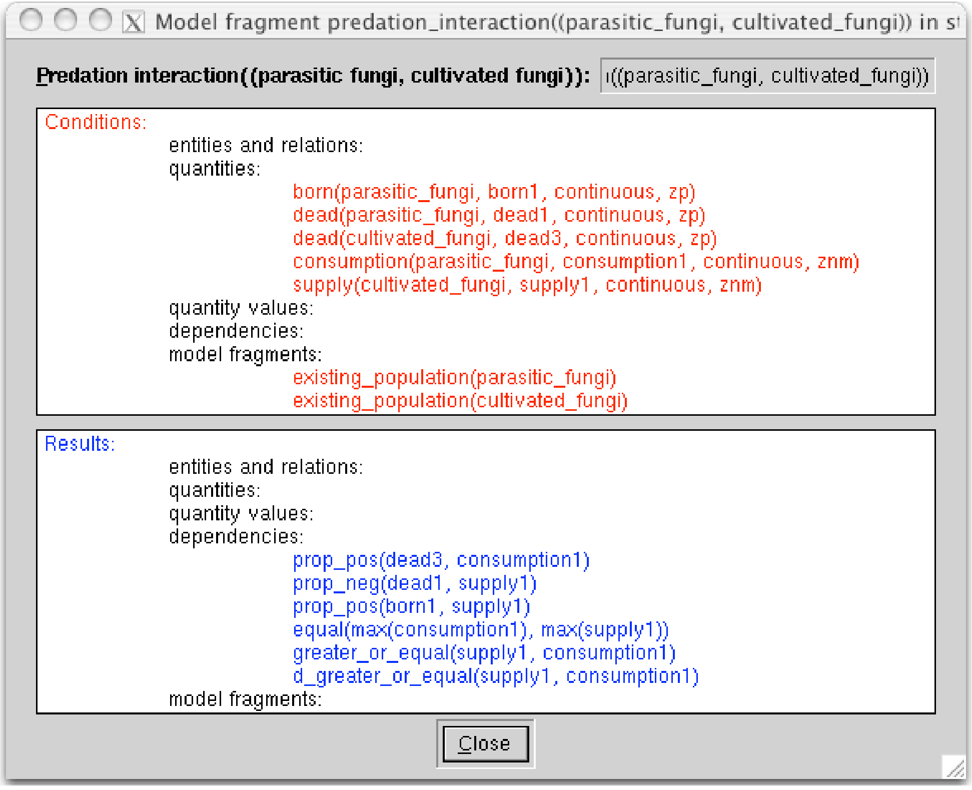

Figure 20: Ants Garden: Predation / Parasitism interactions

6.3 *Model debugging, simulation and finalising*

As the modelling effort progresses and the model becomes more complex, it becomes more difficult to find the cause of problems that hamper the quality of simulations. That is why it is important to take a step-by-step approach to build the model, and carefully understand how the implemented structure produces the behaviour shown in the simulations. The following recommendations are considered important:

- Create the scenarios first and after that start implementing the model fragments. Scenarios should express the general ideas the modeller wants to convey and as such are important starting points for the behaviour generation.
- Be careful to correctly represent the system structure (entities, attributes, and configurations) in the scenario, and to ensure that this structure is correctly repeated in the model fragments.
- For each model fragment implemented, run a simulation and check the results. This way, undesired behaviour can be spotted and fixed as soon as it occurs and, as the model grows in complexity, the modeller can be sure that previous steps are correct.
- Continuously check the behaviour produced by processes. Given that processes start changes in the system, it is wise to be sure about the propagation of these effects throughout the whole set of quantities.

6.3.1 Understanding states

The modeller should inspect the whole state graph and check the results captured in each state of behaviour as produced by the reasoning engine. The relationship between structure and behaviour should be completely understood by the modeller and the combination of qualitative values of all quantities assessed.

The modeller must keep in mind that the reasoning engine produces all the combinations between values that were not prohibited by the model implementation. For example, a particular value of a quantity may be, in principle, associated to all the possible values of another quantity. If one or some of these combinations do not make sense or should not be allowed, constraints must be added to the model (using the modelling primitives). Inspecting the values of all quantities and checking the correctness of combinations is fundamental for model construction and debugging.

6.3.2 State transitions

Finding state transitions is done in three steps: terminations, orderings, and transitions to successors states. Actually, Garp3 allows the user to run the simulation step-by-step, going through these three steps for each state. It is important for a modeller to understand and verify each of these steps in detail, so that the modeller is sure that the engine has found the correct terminations, made the expected orderings, and generated the appropriate successor states. If the results are suboptimal the modeller should change the model details.

Modellers should be aware that state transitions follow general rules embedded in the reasoning engine and the constraints introduced in the model by using modelling primitives. The modeller, in general, does not change the transition rules. Thus, control of state transitions shall be handled by changing the modelling primitives in the model.

6.3.3 Removing branches

Given that qualitative values are not exactly defined, the occurrence of ambiguities is quite common in qualitative models. For example, suppose there are two opposing influences on a quantity and no information about which influence is stronger. In such a situation, the reasoning engine tries all the possibilities (three, in this case). When multiple ambiguities occur in a model, the branching may explode and many states may appear.

There are a number of ways to reduce ambiguities and therefore the number of states produced in the simulation. The following are mentioned here: the use of assumptions, reducing the number of processes, setting the values of certain quantities and using correspondences between specific values or entire quantity spaces.

The use of assumptions is an effective way of removing branches and therefore reducing the complexity of the simulation. For example, it is possible to reduce the number of states if we assume that the derivative of a particular rate takes the same value as the derivative of a state variable[8]. For example, although it may be possible that when the number of individuals in a population is increasing, the birth rate is increasing, stable or decreasing, a modeller may assume that the birth rate derivative follows the derivative of the number of individuals.

Reducing the number of processes is another mechanism to reduce the simulation complexity. This can be done either by replacing rate and state variable (two quantities) by one quantity that is indirectly influenced, that is, affected by a proportionality, or by using assumptions to give a certain perspective to the model. The already mentioned use of the 'closed population' assumption in the Ants' Garden is an example of this approach.

Another way of reducing ambiguities is to settle the value of some quantities under certain conditions. For example, defining that a quantity will only reach a certain value

[8] The Quantity directly influenced by the rate.

when another quantity is greater than a certain value. This can be implemented by means of assumptions or exploring the conditions / consequences parts of model fragments.

Finally, one of the most effective ways of reducing excessive branching is to use correspondences between specific values of quantities or entire quantity spaces. For example, it can be easily understood that when the number of individuals in a population is zero, then both birth rate and death rate have to be zero. This can be implemented by means of a correspondence between values. An example involving the whole quantity space is present in the communicating vessels system between the quantities pressure, height and amount (see e.g. Figure 15).

6.3.4 Finalisation

A model is done when the model fragments library expresses the most relevant concepts and implements features in a clear and organized way. The causal model expresses the most important processes and the propagation of their effects to the whole system. Implemented scenarios explore the model library and produce simulations that show the most interesting phenomena related to the system. The simulations should produce a manageable number of states, and the results show meaningful changes. Finally, the golden standard to consider a model finished is the objectives set for the model. A model is considered good when it meets the objectives set for its use.

7 Model documentation

Goal: Defining requirements for producing quality documentation of the modelling effort, the model, and the simulation results produced by the model.

In this section we are concerned with technical model documentation[9]. The model documentation should give readers sufficiently detailed information, such that the reader in principle can redo the modelling effort and arrive at the same results. In addition, the model documentation should provide the reader with sufficient information concerning the details captured by the model, so that it can be fully understood. In many aspects, model documentation consists of an organised summary of the ideas previously presented in this report (see also Figure 2). In the following sections the main parts of model documentation are discussed.

7.1 Context, model objectives, and use of the model

Model documentation starts with the name of the model. This should summarise what is in the model (for example, 'the communicating vessel model', 'the Ants' Garden model'). Avoid names such as 'test2' and 'general water4'.

The 'orientation and initial specification' step, has produced three outputs that are now available and should be used to introduce the model context, its objectives, and use:

- Documentation (regular text and/or presentation)
- Model goals (& target audience)
- Concept map

Although the modelling effort may have changed the view on some of the initially specified details, it is important to present an account of what was intended and to what extent that was realised. So, it is expected that the model documentation always begins with a (slightly rewritten) version of the output from the 'orientation and initial specification' step.

The objectives to be met by the model and the identification of potential end users are essential in model documentation. Both make it possible to assess model quality and adequacy. The objectives set for the model are the golden standards for model evaluation: a good model is the one that achieves the goals defined for it. Identification of end users make it possible to understand the modeller's choices and to assess the adequacy of the model to its intended use. For example, models for the general public may rely on everyday vocabulary and common sense knowledge, while for university students a more formal approach would be required.

7.2 Global structure and behaviour

The second part of the model documentation consists of the outputs created during the 'system selection and structural model' step and the 'global behaviour' step. In principle the documentation should included previously developed details concerning:

- Global structure
    - System structure
    - System environment and external influences
    - Assumptions concerning structure
- Global behaviour

[9] As for instance opposed to publishing an article in a journal.

  - Processes
  - External influences and deliberate actions
  - Causal model
  - Typical situations and behaviours
  - Assumptions concerning behaviour

As with the previous section, the modelling effort may have changed the view on some of these details. It is therefore important to present an account of what was intended, to what extent that was realised, and why modifications were needed.

7.3 Implementation details

The third part of the model documentation consists of a detailed description of all the model ingredients that constitute the implemented model. In general this can be a listing of all the ingredients created in the software (Section 6) annotated with the details specified in the step 'detailed system structure and behaviour' (Section 5). Specifically it concerns a full description of the:

- Entity hierarchy (use picture generated by Garp3 workbench)
- Attributes and values (create overview table)
- Configurations (create overview table)
- Agent hierarchy (use picture generated by Garp3 workbench)
- Assumptions hierarchy (use picture generated by Garp3 workbench)
- Quantities and Quantity spaces (create overview table)
- Scenarios (use pictures generated by Garp3 workbench)
- Model fragments (use pictures generated by Garp3 workbench)

For many of the ingredients, screenshots of the Garp3 workbench can and should be included. Only for attributes, configurations, quantities and quantity spaces it is more appropriate to create tables listing all the ingredients of these types. In both cases, textual descriptions created during the 'detailed system structure and behaviour' step should be used to explain and discuss the ingredients.

Notice that the model documentation should include a description of *each* model fragment and *each* scenario. When a model fragment has subtypes, it is not necessary to repeat all the issues originating from the super type in the subtype. In such cases the documentation can focus on the newly added features in the subtype model fragments.

7.4 Simulation results

The fourth part of the documentation concerns the simulations results, based on the model. A key result of running a simulation is the state graph, which is the central output of running a scenario. The following details should be described:

- State graph (use picture generated by Garp3 workbench)
  - Scenario (for which the graph was generated)
  - Begin states
  - End states
  - Branching points
  - State transitions (use picture generated by Garp3 workbench)
    - Transition history (use picture generated by Garp3 workbench)
    - Value history (use picture generated by Garp3 workbench)
    - Equation history (use picture generated by Garp3 workbench)
  - Behaviour paths (use picture generated by Garp3 workbench)
  - States
    - Structural model (use picture generated by Garp3 workbench)

- Quantities and values (create overview table or use picture)
- Model fragments (create overview table or use picture)
- Dependency diagram (use picture generated by Garp3 workbench)

For each simulation to be documented, the modeller should include the name of the initial scenario and initial values of quantities, the state graph obtained from that initial scenario, definitions of initial and end states and the most relevant behaviour paths. Analysis of branching in the behaviour graph is useful for understanding ambiguities and to identify where more knowledge is required for a better representation of the system. A discussion about these ambiguities and how to solve them must be included in model documentation.

An overview of the transitions present in a state graph, are shown in the transition history. The details in this history can be modified by selecting a particular subset of states, for instance a set of states that implement a particular behaviour trajectory. The value history and equation history may show details that are also present in the transition history, but the details may also be different. In principle, the value history and equation history show all the values and equations present in the selected states and may also include values and equations that do not change throughout the state graph. The modeller has to decide which of these options (or combination of options) provide the best means to show the required details.

The state graph produced during simulation consists of a set of states. The details in each of these states may be different. Probably not all states can be described fully (because there are too many details). The modeller has to decide upon the subset of states that should be discussed, for instance because they have important features. Typically, this involves the begin states, end states, and branching points. For the selected states the modeller should again decide upon the information to be discussed (because discussing all the details for each state is probably too much and therefore not very realistic). Typically, the dependency diagram should be discussed; it shows how the reasoning engine assembles the model fragments into a full causal model for a given state. Possibly the model fragments active in the state, the structural details to which the state refers and a list of the important quantities, their quantity spaces and their current values (magnitude and derivative).

In general a modeller should keep in mind that readers interested in the model and its simulation results want to learn about as many details as possible. Writing proper model documentation on the simulation results is a difficult task and time consuming. However, it is important, and should not be neglected. Examples of possible listings are shown below.

7.4.1 Running example: The Ants' Garden

The two sections below are taken from Bredeweg and Salles (in press). They are not full descriptions of simulation results. In general, significantly more information should go into the model documentation. However, the sections exemplify the kind of things that should be discussed in this respect.

State graph and value history

Feeding the scenario as shown in Figure 17 to Garp3 results in the simulation show in Figure 22. The state-graph (LHS) has 15 states, from which 9 are end-states (states that do not produce any follow up): [3, 6, 7, 8, 9,10, 12, 13, 14]. The values for the main quantities (Number_of) are enumerated in the value history (RHS). The initial scenario leads to four states [1, 2, 3, 4]. Thus, according to the simulator there are four possible interpretations of this initial situation. In each state the magnitude of Number_of is normal and the states differ on the derivatives calculated for the population sizes. In state 1, all populations decrease. In state 2, all populations increase except the parasitic fungi decreases. In state 3, all populations are steady. In state 4, all populations increase except the parasitic fungi is steady. Following these initial states there are 17 possible behaviours: [3], [1 → 14], [1 → 12], [1 → 15 → 12], etc. The following main behaviours can be found in the state-graph:

- Different ways of coexistence, e.g. [3]
- Complete extinction of the garden, e.g. [1 → 15 → 12]
- Ants, bacteria, and cultivated fungi reaching their maximum size, e.g. [2 → 11 → 7]
- Elimination of the parasitic fungi, e.g. [1 → 10]

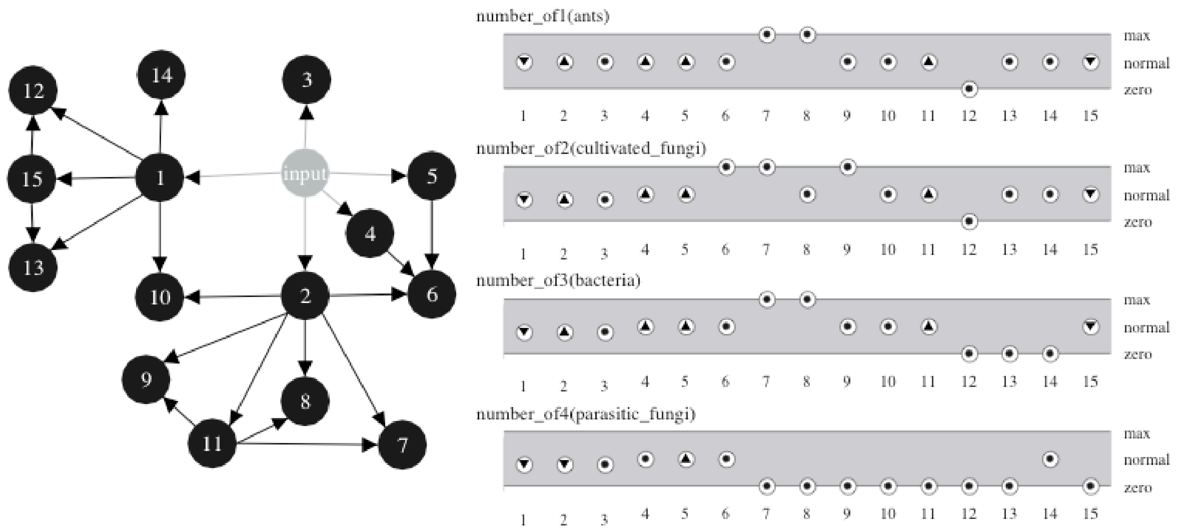

Figure 22: State graph and value history for the Ants' Garden (4 interactions)

Each state has approximately 35 model fragments that specify behavioural details captured by the state. Consider for example state 1. For each single population five model fragments are found: 'population' (defines the structural details of a population, and represents a condition for other population-related model-fragments), 'assume_closed_population' (introduces Born and Dead, and the related indirect causal dependencies, while ignoring migration), 'existing_population' (Number_of > zero, distinguishes it from an extinct population, as for instance in state 12), 'mortality' (introduces the direct negative influence from Dead on Number_of), and 'natality' (introducing the direct positive influence from Born on Number_of). For each interaction type there are at least 3 model fragments, e.g. for symbiosis: 'symbiosis' (defines the structural details for the interaction to become active, introduces the main quantities, and represents a condition for the other model-fragments detailing the interaction), 'symbiosis_interaction' (introduces the causal dependencies that implement the interaction), and 'symbiosis_assumptions' (specifies interaction specific assumptions). With respect to the latter, consider for instance commensalism. To set the strength of the benefit on the affected population we use two versions of commensalism: medium and high impact. Medium impact means that the benefit is partially responsible for changes in the other population. High impact means that changes in the other population are fully determined by the benefit. The 'assumptions' model fragment is used to represent such details.

Dependency diagram

The causal model that is assembled by these applicable model fragments for state 1 is shown in Figure 23. It shows the four populations, each being influenced by its basic processes (Born and Dead), and the interactions between them (2x Benefit, Supply, Consumption, and Pollution). …[10]

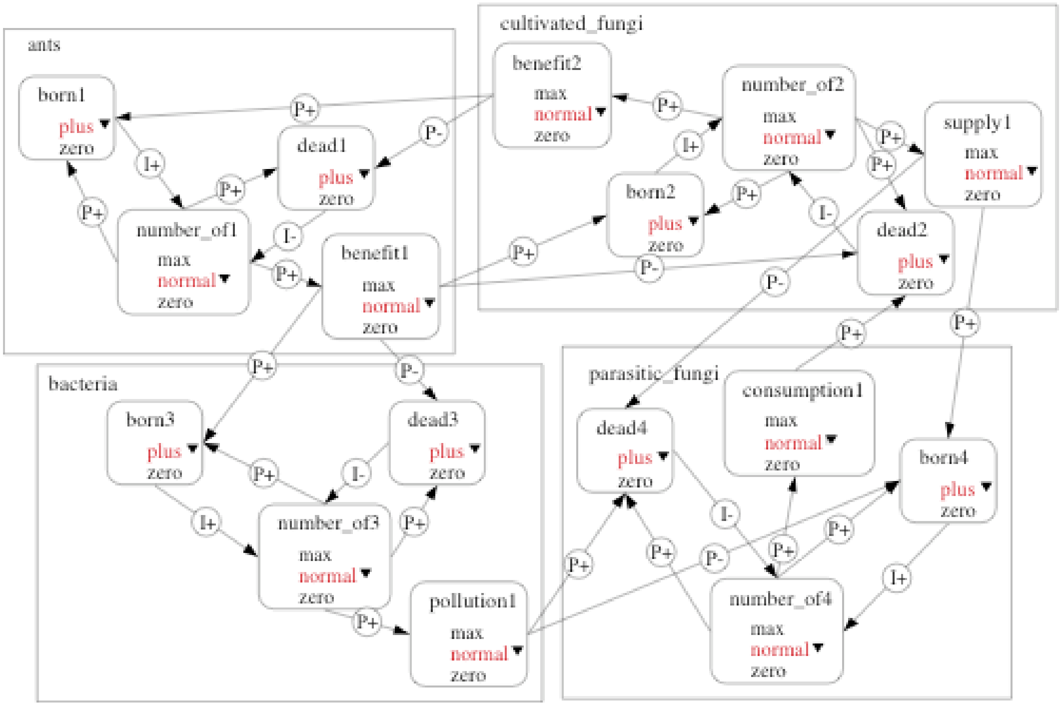

Figure 23: Dependency diagram of the Ants' Garden in state 1 (4 interactions)

7.5 Model evaluation

A model should be validated. The validation methodology used and the results should be included in model documentation, the fifth part of the documentation. Validation starts with references to the available literature. In this case, the results produced by the model should be consistent with other studies found in the literature. This is the minimum to be done in terms of validation. Model documentation may include a text with such comparison, with complete bibliographical references.

It is desirable to evaluate the model with end users and/or domain experts. The modeller may ask them to explore and study the model. Verbal interaction between the modeller and the evaluators, and/or questionnaires, can be used to register the feedback given by the evaluators about the model. A description of the evaluation process and all the material used (interview notes and structure, questionnaires) should be included in the model documentation.

Often models can be used as part of more complex representations or inspire the construction of other models. Information about such applications of the model should be included in model documentation.

[10] Additional text on the dependency diagram is left out, but see Bredeweg & Salles (in press).

7.6 Summary table

The sixth part of the model documentation concerns the summary. The documentation should include tables that summarise the model contents and simulation results. Below the basic outline for such tables is presented in Table 6, 7, 8, 9, 10, 11, and 12.

Table 6: Entity summary

Entity	**Super type**	**Description**[11]
…	…	…

Table 7: Configuration summary

Configuration	**Entity** *(from)*	**Entity** *(to)*	**Description**
…	…	…	…

Table 8: Attribute summary

Attribute	**Entity/Agent**[12]	**Values**	**Description**
…	…	…	…

Table 9: Quantity summary

Quantity	**Entity/Agent**	**Quantity space**	**Description**
…	…	{… , …, …}	…

Table 10: Scenario summary

Scenario name	…
Initial values	…
Initial equations	…
Description	…

Table 11: Model fragment summary

Model fragment name	…
Super type	…
Description	…

Table 12: Simulation summary[13]

Scenario	…
Full simulation	Total nr of states
Begin state(s)	[Nr.], [Nr.], …
End state(s)	[Nr.], [Nr.], …
Behaviour 1[14]	[Nr. → Nr. → …]
Behaviour description	Description of the behaviour
Behaviour 2	[Nr. → Nr. → …]
Behaviour description	Description of the behaviour
…	…
Overall description	Description of the overall simulation

[11] Descriptions should be short, about 10 to 20 words.
[12] List of entities and/or agents that use the attribute.
[13] Create a simulation summary for each simulation.
[14] Use two new rows for each behaviour, e.g., Behaviour 1, Behaviour 2, …, Behaviour n.

7.6.1 Running example: Communicating vessel system

To illustrate the model documentation summary tables, a summary for the communicating vessel system is show below in Table 13, 14, 15, 16, 17, 18, and 19.

Table 13: Communicating vessel system: Entity summary

Entity	**Super type**	**Description**
Substance	Entity	Physical entities that can change state between solid, liquid, and gas.
Liquid	Substance	Substance between freezing and boiling.
Water	Liquid	Liquid of type H_2O.
Oil	Liquid	…
Object	Entity	Rigid entities.
Pipe	Object	Facilitates flow of substances.
Container	Object	Can contain substances.

Table 14: Communicating vessel system: Configuration summary

Configuration	**Entity** *(from)*	**Entity** *(to)*	**Description**
Contains	Container	Substance	Specify which containers can contain which substances.
From	Container	Pipe	Specifies direction of substance flow.
To	Pipe	Container	Specifies direction of substance flow.
Connected			Not used in the model.

Table 15: Communicating vessel system: Attribute summary

Attribute	**Entity/Agent**	**Values**	**Description**
None			

Table 16: Communicating vessel system: Quantity summary

Quantity	**Entity/Agent**	**Quantity space**	**Description**
Amount	Liquid	{Zero, Plus, Max}	The total amount of a substance.
Height	Liquid	{Zero, Plus, Max}	Height of a liquid column.
Pressure	Liquid	{Zero, Plus, Max}	Pressure bottom of a liquid column.
Flow	Pipe	{Min, Zero, Plus}	Liquid flow through pipes

Table 17: Communicating vessel system: Scenario summary

Scenario name	Both containers partially filled, left column is higher.
Initial values	Height(Container left)={plus} Height(Container right)={plus}
Initial equations	Height(Container left) > Height(Container right)
Description	A flow is expected from left to right until the heights of the two liquid columns equalise.

Table 18: Communicating vessel system: Model fragment summary

Model fragment name	Contained liquid
Super type	Static
Description	Introduces amount, height and pressure and their dependences in terms of proportionalities, correspondences, and (in-)equalities.

Model fragment name	Liquid flow
Super type	Process
Description	Specifies the details concerning liquid flow between two connected containers based on pressure differences.

Table 19: Communicating vessel system: Simulation summary

Scenario	Both containers partially filled, left column is higher.
Full simulation	4 states
Begin state(s)	[1]
End state(s)	[2], [3]
Behaviour 1	[1 → 3]
Behaviour description	This behaviour represents the expected behaviour: unequal heights become equal in state [3].
Behaviour 2	[1 → 3]
Behaviour description	This behaviour is similar to behaviour 1. The difference is that the container on the right ends up being fully filled. The length of the column apparently equals the length of the container. The latter stays implicit in the model, because container height is not represented explicitly.
Behaviour 3	[1 → 4→ 3]
Behaviour description	This behaviour ends similar as behaviour 2. However, in state [2] the right container overflows. Apparently liquid column in the right container was higher then the height of the container on the right.
Overall description	The behaviour of the communicating vessels turned out to be more complex then initially anticipated. Two unexpected states appeared in the simulation (both representing correct behaviour). The additional behaviours are the result of missing information about the length of the containers and how that compares to the height of the two liquid columns. Thus, to better represent the behaviour of the communicating vessel system the length of the containers should also be modelled.

8 Vocabulary standards and conventions

Goal: *Defining standards and conventions so that the results created by modellers are easier to understand by others.*

Communication about the qualitative models is essential for the NaturNet-Redime project. It is therefore important to adopt a set of conventions to be used for describing modelling primitives and simulations. A summary of conventions is shown in Table 20.

Table 20: Summary of conventions

Convention	**Example**
Use everyday language to name entities, quantities and quantity values; avoid using abbreviations or *cryptic* words.	The entity *Soil* has quantity *Fertility*, with possible qualitative values *low*, *medium*, *high.*
Use italics to refer to modelling primitives (entities, quantities, configurations, MF names etc.) so that they can be easily recognized in the text.	The MF *Soil fertility* describes the relations between quantities *Fertility* and *Crop production*, so that P+(*Crop production, Fertility*).
Use capital letters for representing entities and quantities.	The entity *Industry* has quantity *Pollution*, so that *Industry*: *Pollution.*
Do not use plural names for entities or quantities.	For example, avoid names such as *Trees* and *Numbers.*
Do not use numbers to name entities or quantities.	For example, avoid names such as *Concentration1,* or *Rate1* and *Rate2.*
Avoid single letters, abbreviations and composed words for naming entities and quantities.	For example, avoid names such as entity *E* has quantity *N*.
There is no need for naming quantities with the name of the entity to which it belongs: Garp3 does it automatically.	For example, avoid names such as *Biomass_of_tree*
Prefer everyday words to describe quantity values.	For example, *large*, *increasing*, *short*, *dry*, *hot.*
Create names for quantity spaces using the first letter of each qualitative value.	For example, mzp, zlmh, or p.
Present the set of possible qualitative values in a quantity space between curly brackets.	For example, {minus, zero, plus}; {zero, low, medium, high}; {plus}.
Use numbers between flat brackets to represent states.	For example, [7] or [1,3,5].
Put numbers and arrows between flat brackets to represent a behaviour path.	For example, [3 → 5 → 4]
Use I and P to represent, respectively, direct influences put by processes and propagation of changes via proportionalities (indirect influences).	For example, I+(Concentration, Emission) P+(Disease, Concentration)
Use the traditional notation to represent inequalities.	For example, *Concentration*(*Water*) < *Concentration* (*Air*)
Use correspondences to establish simultaneity of two specific quantity values.	QS_Pollution {large} corresponds to QS_Oxygen {small}.

General remarks

As a general remark, when writing documents it is necessary to draw the reader's attention to the modelling primitives in the text. As the vocabulary used in qualitative models is close to everyday language, it is important to use *italics* to make clear when a particular word is part of the model. For example, 'the entity *Tree* has a quantity *Biomass* to represent tree biomass'.

The use of capital letters to represent entities or quantities may be confusing. In Prolog, the programming language used to implement Garp3, capital letters are used to identify variables. Once they are instantiated, the instance is identified with lower case. As a rule, avoid the use capital letters in the middle of terms. Also do not use numbers to identify entities or quantities. Because this may interfere with low level details of the Garp3 reasoning engine and produce confusing results, such as *Substance1*: *Concentration1* and *Substance2*: *Concentration2*, becoming: *substance11*, *substance21*, *concentration11* and *concentration21*.

Entity (E)
In order to represent physical objects or abstract concepts included in the model as entities, it is recommended to select well known words: for example, *River*, *Tree*, *Population*. Avoid using abbreviations and composed words, for instance, *Mamm* (for mammals) or *Verteb_pop* (for vertebrate population). Do not use the plural form of the name of the entity: use *Container*, *Forest*, and *Substance* instead of *Containers*, *Forests*, and *Substances*.

Quantity (Q)
Quantities are important elements in qualitative models, because they represent the most relevant properties of the entities of interest in the model. Quantities should be named in a way that everybody can understand. For example, *Concentration*, *Number_of*. Avoid as much as possible the use of single letters, abbreviations and composed words (e.g. *N*, *Conc*, *Nitrateconcentration*).

E and Q Association
In a formal representation of quantities, show the entity associated to each quantity. Note that this association is represented in Garp3 by using a colon, as in *E: Q*. For example, *Population*: *Number_of*, *Substance*: *Concentration*.

In a number of models, the same quantity is associated with different entities. Suppose that a model includes the entities *Nitrogen* and *Phosphorus* and for both the modeller wants to refer to the quantity *Concentration*. As Garp3 associates each quantity to an entity, even when different entities have quantities with similar names it is possible to identify the pair *E: Q*. In the given example, Garp3 creates the associations *Nitrogen: Concentration* and *Phosphorus: Concentration*.

Therefore, there is no need for naming quantities with the name of the entity to which it belongs. Thus, avoid names like *Nitrogenconcentration* or *P-concentration*. Actually, it is recommended to keep the same quantity linked to different entities. For example, in a model the quantity *Temperature* may be associated to different quantities.

Quantity value
Each quantity value represents a qualitative state of a particular entity. Quantity values have two components, namely a magnitude and a derivative. The magnitude is the current value of a quantity and the latter shows the direction of change. Communication of magnitude values has to be easily understood. Thus, values should be identified by well-known words: *small*, *normal*. The derivative can assume three values: negative, zero, positive. There values read that the quantity is decreasing, stable, and increasing.

All in all, quantity values should be represented as a pair between angle brackets as follows: <*magnitude*, *derivative*>. Examples: <*normal*, +>; <*hot*, *0*>; <*large*,–>

Quantity Space (QS)
The possible qualitative values a quantity may assume in a particular qualitative model are represented as a set named Quantity Space (QS). They are identified by a name and an ordered set of qualitative values.

Although necessary for model implementation in Garp3, the QS name is not of interest for the public. It is actually a label, often created out of the first letter of each qualitative value. For example, *mzp* (for minus, zero, plus); *zp* (for zero, plus); *lmh* (for low, medium, high). Sometimes, information about QS can be communicated in phrases such as 'the quantity biomass *Growth rate* has QS mzp'. More useful is to represent a QS as a list of possible values between brackets after the Q name: *growth rate* QS = {minus, zero, plus}. Other QS examples: {*small*, *medium*, *large*} and {*below_normal*, *normal*, *above_normal*}.

Derivatives have only one QS, namely QS = {min, zero, plus}.

In Garp3, quantity spaces always have points and intervals. A complete description of a QS should then be {point(zero), interval(low), point(medium), interval(high), point(maximum)}. In general, this is not represented and we refer to the same QS as {zero, low, medium, high, maximum}.

Qualitative state
Each qualitative state can be described by its value and (in)equality statements. In Garp3 numbers identify states in the behaviour graph. Textual references to states should use numbers between flat brackets: [7]. To represent two or more states, use commas: [7,13,21]

Behaviour path
A behaviour path is a sequence of successive qualitative states, connected by transitions. To represent a behaviour path, use numbers and arrows between flat brackets. For example: [7 → 8 → 9→10].

Direct influences and qualitative proportionalities
These primitives should be represented as they traditionally appear in the literature: using I and P followed by the sign (positive or negative) and the elements open bracket, influenced quantity, comma, influencing quantity, close bracket, in that order.

I+(state variable, rate); I– (state variable, rate)

Examples: I+(*number_of*, *birth_rate*); I– (*number_of*, *death_rate*)

P+(influenced quantity, influencing quantity);
P–(influenced quantity, influencing quantity)

Examples: P+(*occupied area*, *number_of*); P–(*available space*, *number_of*)

Note that, although qualitative proportionalities are sometimes represented with the Greek letter alpha, we use the letter P.

Inequalities

In order to represent inequalities between magnitudes, the project will use traditional notations: A > B; A ≥ B; A = B; A ≤ B; A < B.

For example:
birth_rate > *death_rate*; *birth_rate* >= *death_rate*; *birth_rate* = *death_rate*.

Garp3 allows for defining inequalities between derivatives. They are represented as '∂': ∂A > ∂B; ∂A ≥ ∂B; ∂A = ∂B; ∂A ≤ ∂B; ∂A < ∂B.

For example: *∂birth_rate* = *∂death_rate*[15].

[15] Notice that in the software these are depicted as: A ∂> B; A ∂≥ B, A ∂= B, A ∂≥ B, A ∂> B.

9 Conclusion

This document provides a framework to support and organize the capture of conceptual knowledge about systems and their behaviour using a qualitative reasoning approach. This framework defines a methodology to be used for developing a conceptual understanding of each case study to be carried out within the Naturnet-Redime project in work package 6. The use of a structured approach assures that results can be compared more easily and assessed on a similar basis. The basic organisation of the presented approach is as follows:

- Initial orientation on how to start the modelling effort, establishing what is to be modelled, reasons for representing phenomena in a qualitative model, objectives to be met with the use of the model and identification of end users. The use of concept maps and different ways of presenting issues of interest is recommended in this initial specification phase.
- Identification of the target system structure and its constituents. Defining the system structure requires the establishment of boundaries and of the system environment, from where external influences may affect the system. Justification of modelling decisions concerning what was included and what was left out of the model, in order to explicate the relevant structural assumptions.
- Description of the behaviour to be captured by the model, first at a general level of abstraction. In this phase, some elements are very important: identification of the most relevant processes, external influences and deliberate actions; design of a causal model to represent how changes start and propagate to the rest of the system; specification of interesting initial scenarios for simulations and what is expected as behaviour graphs and transitions between states; it is also important to make explicit representations of the relevant modelling assumptions concerning the system behaviour.
- Detailed description of the system structure and behaviour, in order to guide the definition of the model ingredients. In this phase, important modelling decisions are made, because the global analysis of the system structure and behaviour from the previous phases is translated into the qualitative reasoning modelling approach and accompanying vocabulary. The definition of entities, attributes, and configurations provide the required structural details; agents are useful for representing external influences, if they exist; assumptions referring to structural and behavioural aspects are designed for implementation; quantities and quantity spaces specify relevant properties and their qualitative states; details of what should be included in initial scenarios are defined with reference to the representations of system behaviour; finally the specification of conditions for model fragments to become active and the consequences they introduce in the model provide a plan for the implementation phase.
- Implementation is the phase in which the modeller actually creates a model using Garp3. This phase in general requires debugging and small adjustments in the model implementation, such that each of the scenarios produces the desired behaviour when simulating the model.

Finally, the framework includes guidelines for model documentation and a common vocabulary for communicating modelling decisions, assumptions, model details, and simulation results obtained from the models.

10 Literature

Bredeweg, B., Salles, P. and Neumann, M. (in press) Ecological Applications of Qualitative Reasoning. In: Ecological Informatics. Understanding Ecology by Biologically-Inspired Computation. Recknagel, F. (ed.) (2nd version).

Bredeweg, B. and Salles, P. (in press) The Ants' Garden: Complex interactions between populations and the scalability of qualitative models, AI Communications.

Cañas, A.J., Coffey, J.W., Carnot, M.J., Feltovich, P., Hoffman, R.R., Feltovich, J., Novak, J.D. (2003) A summary of literature pertaining to the use of concept mapping techniques and technologies for education and performance support, Technical report, The Institute for Human and Machine Cognition, Pensacola FL, USA, http://www.ihmc.us (visited August 31st, 2005).

Currie, R.C., Scott, J.A., Summerbell, R.C. and Malloch, D. (1999a) Fungus-growing ants use antibiotic-producing bacteria to control garden parasites, Nature 398, 701-704.

Currie, R.C., Mueller, U.G. and Malloch, D. (1999b) The agricultural pathology of ant fungus gardens, Proc. Natl. Acad. Sci. USA 96(14), 7998-8002.

Fell, (1996) New Scientist, Vol. 151, Issue 2041, page 19.

Lumpkin, S. and Hsia, S. (2004) The First Farmers. Web-article on the Smithsonian National Zoological Park's Zoogoer site http://nationalzoo.si.edu/Publications/ZooGoer/2004/4/antfarmers.cfm

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

Salles, P. and Bredeweg, B. (2003). A case study of collaborative modelling: building qualitative models in ecology. *Artificial Intelligence in Education: Shaping the Future of Learning through Intelligent Technologies*, U. Hoppe, F. Verdejo, and J. Kay (eds), pages 245-252, IOS-Press/Ohmsha, Japan, Osaka.

Salles, P., Bredeweg, B., and Bensusan, N. (in press) The Ants' Garden: Qualitative models of complex interactions between populations, Ecological modelling.