Model Minimization in Qualitative Reasoning

Andreas van Cranenburgh
0440949
acranenb@science.uva.nl

Hanne Nijhuis
0364568
hnijhuis@science.uva.nl

June 3, 2010

Abstract

Qualitative Process theory provides a way of modeling causality in a system on a qualitative level. Previous work has attempted to automatically induce such qualitative models from behavior. We extend this work by turning the resulting monolithic models into minimized, modular models, which makes the induced models re-usable. The resulting minimization algorithm is general enough to apply to user created models as well. We also sketch other possible improvements for model induction, based on the exploitation of human intuitions about systems rather than the current reliance on complete and error-free data.

Contents

1 Introduction

Qualitative Reasoning is the field of research that concerns modeling conceptual knowledge. Many approaches have been taken to formalize this knowledge and in recent years a lot of successful appliance has been shown. The modeling however is a complex and time consuming task, so an automated modeling algorithm is desirable. Although the basis for this algorithm has been successfully defined by Buisman, and later extended by Liem and Van Weelden, there are still some things missing to making this algorithm usable in practice. The current approach makes strong assumptions about the input, which limits the applicability. Also, the output of this automated modeling is a monolithic (single, often large) model fragment. In qualitative reasoning, the re-usability of fragments is an important feature, so in this paper we introduce a procedure which can divide the monolithic fragment into smaller, reusable fragments which are easier to comprehend.

We start with a literature review in section 2, then in section 3 we will introduce the theory on which our algorithm is based. The approach itself is explained in section 4 and the results are shown in section 5. Finally, section 6 and 7 end with a discussion and conclusion.

2 Literature Review

We will start with a literature review as an introduction to the field and to supply insight in approaches that have been attempted. In this review we will cover papers on general Qualitative Reasoning (QR) topics and also on the Automated Modeling (AM) topics.

2.1 Qualitative Reasoning

Qualitative Reasoning is an approach from Artificial Intelligence that provides means to express conceptual knowledge such as the physical structure, causality, processes, etc. This enables reasoning about phenomena for which numerical information is sparse or missing, or when such information is too complex to grasp. The following papers describe the foundations of qualitative reasoning.

2.1.1 A qualitative Physics based on confluences

De Kleer et al. [6] present a framework for formalizing physics in a manner that is *qualitative* rather than *quantitative*. Whereas physicists describe the world in terms of continuous differential equations, the rest of the world uses an implicit psychological model of naive physics. The framework of de Kleer et al. falls between these two extremes of precision and informality. They employ confluences, which are qualitative differential equations. Although their framework is completely qualitative, as opposed to standard physics, on the other hand their framework is formal in the sense of explicitly defining causal and structural relations, as opposed to the psychological models of humans (presumably). Moreover, their approach is based on deriving qualitative physics from first principles, like normal physics, as opposed to the contextual and tacit knowledge of mere mortals.

This approach is presented as being useful for physics education, expert systems and even physics, in that it explicitly deals with causality. The latter is in contrast with standard physics, in which laws are merely exceptionless correlations with predictive value. Although the mathematics behind physics is relatively formal, doing physics depends on a pretheoretical understanding of the world, of which there is no successful account to date (neither in physics nor in psychology). Textbooks for physics and second language learners clearly contain enough information for humans to grasp such subjects (albeit only after intensive study), however, they are nowhere formal (explicit) enough to be useful for the current computational paradigms; they seem to rely on an extensive common ground.

Naive physics attempts the herculean[1] task of explicitly formalizing all the required knowledge to derive fragments of modern physics in a qualitative manner. The fundamental question at stake here is how deep the rabbit hole goes in terms of the required background knowledge which is so much taken for granted by mortals.

A few assumptions of their approach characterize their methodology:

No function in structure The functioning of the whole should not be encoded in its parts. For example, a light switch should not state that it directly determines whether there will be light, because other things such as the state of the light bulb and the payment of bills also influence such matters, let alone the need for a closed circuit.

Class-wide assumptions Modeling always happens at a certain resolution yielding a level of description. Brownian motion, quantum entanglement and other quaint microcosmic phenomena can be safely ignored not just for certain models, but for any model which aims for the same predictive power as human folk physics. The modeling granularity also includes time and spatial resolution; for example oscillation phenomena are ignored as long as the system needs only a certain negligible amount of time to settle.

Locality This principle states that the behavior of parts cannot influence other parts, except when they are structurally connected. This is strongly related to the focus on causality, and implies that without a locus of causality there can be no model (e.g., an emergent phenomenon like a tornado could not be modeled).

The importance of principles Violating the no-function-in-structure principle has no result on the success of accurately modeling certain behavior, but it does result in ad-hoc models with little value for understanding actual physics. This is comparable to the situation of observing a mere correlation without positing a plausible causal mechanism; this provides no information whatsoever about possible causality and does not further understanding – it is a pointless exercise in data collection.

This fine point is easily glossed over when the goal of deriving first principles fades out of sight; without principles hacking together

[1] or foolish, depending on your disposition

models is pointless, because one will simply continue until the desired output is there, but if the desired output was clear to begin with, it is impossible for a model to add any value.

Advantages of this approach are that it works from first principles, as such it is grounded in science. Other advantages are that the models can be induced from data, even when the latter contains noise. Disadvantages are that these constraint models are less explanatory than process models ([4], p. 110), which we will review promptly.

2.1.2 Qualitative Process Theory

Around the same time as de Kleer & Brown, another approach was introduced by Forbus [4]. The important points of his work are, among others, the *quantity-spaces*, notion of processes and the related *sole mechanism assumption*.

The principle of quantity spaces represents values by a set of ordinal relationships. This enables the system to easily compare different quantities in terms of smaller, equal or larger. If such comparisons would only occur between two values, then sign values would suffice (-,0,+), but for many cases, the ordinal representation is more natural.

The most salient difference with earlier qualitative approaches is the centrality of processes. Examples of processes are flow, motion and phase changes. Physical processes are viewed as the mechanisms through which change occurs. To this end Forbus introduced the *sole mechanism assumption* which states that all behavior must be explained by processes: the direct or indirect effects of some collection of physical processes.

2.1.3 Conceptual knowledge & the Garp3 workbench

Capturing conceptual models can show the consequences of what we believe to be true [2], analogous to formal logic. Using computer simulations large models can be evaluated, showing the interactions of fragments of knowledge. The rest of the text [2] deals with Garp3 which is discussed in more depth in Bredeweg & Salles [3].

Garp3[3] is an implementation of Qualitative Process Theory with a focus on educational uses.

Tools are being build to take a graphical approach to building qualitative models. With the introduction of Garp3, Bredeweg et al. want to preserve the full expressiveness of the QR formalism and to address domain experts and support them in articulating and capturing their conceptual knowledge. The software uses a diagrammatic approach for representing model content.

In Qualitative Reasoning, quantities that describe the dynamic features of a system typically hold qualitative information concerning the current magnitude and direction of change, using an interval scale, consisting of an ordered set of labels, e.g. zero, low, medium, high. such a set of labels is called a quantity space. Landmarks are specific labels within this set (actually points) that refer to situations in which the behavior of the system changes significantly. For instance, a substance reaching its

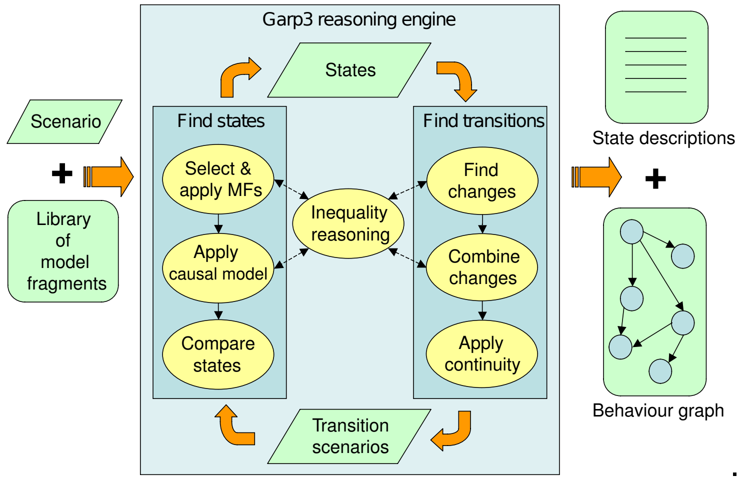

Figure 1: Overview of the Garp3 reasoning engine

boiling temperature will stop getting hotter and start boiling. In qualitative simulations, the behavior of the system is represented as a graph of states that reflect qualitatively distinct system behavior. There is a set of dependencies that capture cause-effect relationships between quantities, defined in such a way that they are both closely resembling the conceptual notions of human reasoning, as well as allowing automated computation because they are grounded in mathematical formalisms. Two typical dependencies are direct influences and proportionalities.

A QR engine has inference mechanisms to assemble the appropriate set of dependencies that describes a system in a certain state of behavior. Two aspects are important for this approach: behavior can be inferred from the physical structure of the system, and knowledge about system behavior is stored in small fragments with conditional information detailing when such a fragment is applicable.

Simulation is done by the Garp3 reasoning engine, using a state-by-state strategy. The *find-states* algorithm constructs initial nodes in the state-graph based on the scenario and model fragments. Then the *find-transitions* algorithm determines all possible changes which form transition scenarios. These are inputted into the find-states procedure again. See figure 1 for an overview.

The reasoning process relies heavily on inequality reasoning. Inequality relations typically change during the simulation, thereby representing the dynamic aspects of the modeled system. Points in a quantity space are called landmarks and the intervals between them are defined by these landmarks. The core inequality reasoning only considers these landmarks and not the intervals. Inequalities can specify the relation between several types of model ingredients. The following five types of inequality relations are used in Garp3:

- *Quantity* $\leftrightarrow$ *Quantity*
- *Quantity* $\leftrightarrow$ *Landmark*
- *Landmark* $\leftrightarrow$ *Landmark*
- *Derivative* $\leftrightarrow$ *Derivative*
- *Derivative* $\leftrightarrow$ *Zero*

At the user-level representation, quantity values, landmarks and derivatives are used, while at mathematical level, only landmarks exist and are used to describe the others (for both magnitudes and derivative quantity spaces).

All the relations in the internal mathematical model of Garp3 have the following form: $rel(Sum_1, Sum_2)$ with $rel \in \{>, \geq, =\}$. The following three basic principles are used to make inferences in the Garp3 inequality reasoning engine:

- Basic algebraic simplification
- Anti-symmetry
- Transitivity

For example:

- $A > B$, $B > C$ (given)
- $(A > B)$ & $(B > C) \rightarrow A + B > B + C$ (transitivity)
- $A + B > B + C \rightarrow A > C$ (simplification)

Garp3 does not discriminate explicitly between inequality statements that always hold (e.g., *inflow* - *outflow* = *net-flow*), and those that may change during simulation (e.g. $T_x < T_x boil \rightarrow T_x = T_x boil$).

2nd order derivatives are important for sound simulations, but complete validity cannot be guaranteed. This issue is in principle unsolvable, but does not form a problem for the level of detail of models built in Garp3.

Future work will focus mainly on improving usability of the software in terms of GUI and debugging options.

Bredeweg et al. conclude that the Garp3 workbench offers easy access to sophisticated qualitative simulation software, providing users the possibility to use QR technology without having to understand low-level implementation details of such automated reasoners. An increasing number of domain experts are using the workbench to capture qualitative knowledge of system dynamics.

2.1.4 Algal bloom in the Danube Delta

Cioaca et al. describe a complex case of modeling a real world system in Garp3. The paper mainly describes the model and the real-world specifics in detail, but for the literature review, after a short introduction, we will focus on the qualitative reasoning aspects that were encountered.

The model is a formalization of algal bloom in the Danube Delta Biosphere Reserve (DDBR). In this world heritage site, the biodiversity is threatened by pollution from two different sources. Heavy nutrient loads from agricultural fertilizers and heavy metals from industry. These forms of pollution influence several conditions of several layers in the water, which in term influence the increase and decrease in size of populations of several species.

The main species concerned in the model are the Diatoms (which compose the phytoplankton algae), the blue green algae (bacterial species called Cyanobacteria), Zooplankton and fish. There are also two agents, *land* and *farming*, which should be considered as influences on the system from the outside world. The quantities used are: *Biomass*, *Average temperature*, *Cover*, *Cyanotoxins*, *Water temperature*, *Light*, *Mortality rate*, *Nutrients*, *Nutrient runoff*, *Dissolved oxygen*, *Production rate* and *Carrying capacity*.

An interesting detail is the way they chose to implement the quantity space for temperature. Since every species has a different optimal temperature range, and they specified one quantity for the water temperature, they have to combine all these optimal ranges and their overlaps in one quantity space. This resulted in a quantity space with 6 landmarks (and 5 corresponding intervals). It is not always obvious how to define these quantity spaces however, because the temperature ranges for these species are not strictly defined by exact boundaries. This is something that often leaves the modeler with ambiguous choices.

The complexity of the system becomes clear when a full simulation (containing all entities and using the most detailed mechanisms and views) is run. This results in a valid (stable single end state), but very large state graph, consisting of 3126 states. This is however not a very useful simulation since such a large number of states makes it difficult to comprehend what is happening in the system. Because of this the authors have chosen five scenarios in which they focus on particular concepts. In these five scenarios they were able to thoroughly investigate their model and actually draw conclusions on how to respond to certain situations in the real world scenario of the DDBR.

The discussion in this paper is elaborate and covers a lot of the problems and ambiguity in the choices that were made. Part of their concern is about modeling the two pollutants in one model. This is possible but results in very large state-graphs. And since the two pollutants seem to have independent effects, they chose not to include them both. This last point obviously brings to mind a more general discussion about the usefulness of qualitative reasoning with respect to complex systems like these. This discussion will be mentioned in more detail in the next section.

2.1.5 Discussion

The idea that Qualitative Reasoning can formalize actual common sense knowledge and human understanding of physics is by now outdated. It is an instance of the "Good Old Fashioned Artificial Intelligence" (GOFAI) approach, [15] and in line with the Symbol System hypothesis of

Newell and Simon [1][2]. Both of these have either failed spectacularly or faded silently, often having been replaced by stochastic and data-driven approaches [13]. Claiming that Qualitative Reasoning can provide a way to model actual common sense knowledge or mental models is ostensively wrong – the list of mismatches with human intuitions is unmanageable, and among these mismatches are the parts where Folk Physics is ostensively *wrong*. Specifically philosophers contend that common sense is holistic at its core [16], whereas explicit, knowledge-based approaches are reductionistic in nature, because their aim is to isolate context-free fragments.

In the introduction of Bredeweg & Salles [2] it is claimed that:

> "Qualitative models excel when theoretical background on the target system is weak, when the problems are ill-defined, and when data are incomplete." ([2], p. 352)

This statement is unfortunately not backed up by *any* supporting evidence. Phrased like this it would seem that qualitative models provide us with a brave new kind of science, which is in fact false, unless one trivializes the claim to "scientists implicitly use qualitative models to do science." What in fact would seem to be the case is that qualitative models provide a formalization of knowledge, and as such, require that this knowledge is perfect: complete, consistent and enumerable; otherwise formalization will grind to a halt immediately, because resolving inconsistency already presupposes a proper understanding. The way humans deal with inconsistency is an interesting topic in itself, but suffice to say that the quoted passage is most likely very far from the truth.

Therefore we submit that it is no longer useful to present QR as a part of Artificial Intelligence or having anything to do with common sense; while QR models are certainly artificial, they are not in the least intelligent, nor are they common or do they make sense to non-experts. They are simply an explicit formalization of a certain set of behaviors of interest. This does not mean that Qualitative Reasoning is itself a fruitless enterprise – it can be used as an educational tool, where there is no need for bold philosophical claims.

With this educational focus in mind it makes sense to try to facilitate the modeling process for non-experts for which QR is useful. To this end we turn to the topic of model induction.

2.2 Automated Modeling

To introduce the recent developments in the field of automated modeling we will now review a few papers on this topic.

2.2.1 Automated modeling in process-based qualitative reasoning

The concept of automated modeling for qualitative process theory was introduced by Buisman [8] in 2007. He motivates the research with the

[2] "A physical symbol system has the necessary and sufficient means of general intelligent action."

goal to relieve the strain placed on experts and beginners, and to speed up the modeling process. The approach he took however, does not attain these goals yet, but is more of an exploration of the possibilities of using Artificial Intelligence to induce models based on behavior graphs. We will give a short overview of the algorithm he developed.

The input for the algorithm consists of:

- Behavior graph (states and state transitions) of a full envisionment
- Scenario (partial information about structural relations between entities and their quantities)
- ISA-hierarchy (full description of entity type hierarchy and quantities)

With this input, the algorithm produces models which are ready for simulation. For each scenario they should give the same output as the original model.

There are certain constraints on the input. The *full envisionment* requirement means that the behavior is known for all possible initial values for all quantities in the system. Also, the derivatives and amounts have to be defined for every state. Finally, the input data cannot contain any noise, i.e., it should be perfect.

The algorithm starts with searching for so called *naive dependencies*, which are dependencies that are consistent with the entire state-graph. They are found by applying consistency rules on all pairs of quantities. These consistency rules consider the amount and derivative values of the quantities for all states, so as to induce possible dependencies. For example, a positive influence between Q_1 and Q_2 could exist if for every state $A_s(Q_1) = D_s(Q_2)$. There will be redundancy in the set of naive dependencies which has to be pruned by filtering out substitutionary groups, which are defined as mutually exclusive possible disambiguations.

3 Theory

In the field of QR, models consist of several smaller fragments which can be roughly divided in three categories: static, process and agent. This categorization is not strict, but gives a certain view on how things work. Static fragments are used to describe the structure of a model, as well as proportionalities between quantities. A static fragment cannot have any agents or influences in it. A *process* fragment should have at least one direct influence and is not restricted in terms of dependencies. Finally, *agent* fragments should be used to describe any influences on a system that are external, hence not part of the system itself.

The importance of fragments and compositionality is underscored by Forbus [5] (emphasis in the original):

> A missing ingredient in early attempts to formalize physical reasoning (cf. [...]) was the idea of *compositionality*. That is a major factor in the flexibility of human reasoning about complex physical systems comes from the ability to use of partial information and combine it as available.

Fragments can be useful to make the complete system more comprehensible – i.e., they are a form of chunking. Also, while building these complex models, fragments allow the user to focus on what happens in a small part of the model, without having to worry about the other 'components'. For example in a model describing a population, separate fragments could describe processes like birth, death, emigration, etc.

Recent work on automated modeling [8, 9, 10] has focused on generating a correct model based on a full-envisionment behavior graph. The current algorithms always produce a single model fragment which describes the complete system. The implicit compositionality of a system is not revealed and comprehending the output could become quite difficult with large models. We therefore introduce a way of splitting and minimizing such monolithic models into smaller fragments while preserving the exact same behavior.

See figure 2 for an overview.

3.1 Interdisciplinary parallels

3.1.1 Finite-State automaton minimization, compression

In automata theory there are well-known results regarding the minimization of grammars. For example there exists an algorithm to obtain a minimized version of a Finite-State automaton (FSA) which generates the exact same language with a minimum amount of states [12].

Naturally this problem gets more difficult higher up in the Chomsky hierarchy; no general purpose algorithms exist. It would therefore be an important result if it turned out that it is possible to encode (a subset of) qualitative model simulation as a Finite-State automaton that generates a regular language containing state-strings corresponding to behavior paths.

A related but less hopeful result is that of Kolmogorov complexity, which states that given an arbitrary string, it is undecidable to verify

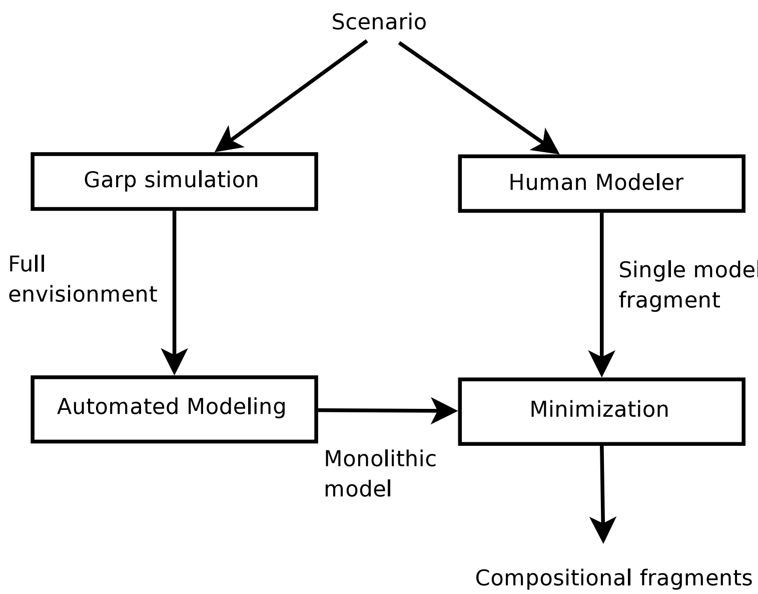

Figure 2: The pipeline of operations and relation to previous work.

whether a given program generating that string is the shortest one to do so (i.e., highest compression), let alone to produce such a program given data.

Determining the exact status of qualitative reasoning with respect to established results in automata theory or formal logic would shed light on the precise complexity and feasibility of model induction.

3.1.2 Minimum Description Length principle

In Machine Learning there is the Minimum Description Length principle (MDL; [11], p. 171, section 6.6), which provides an inductive bias to choose among possible hypotheses for data, given a certain representation (the latter is crucial to success). This provides a powerful heuristic because whenever a hypothesis is found that fits the data, all larger hypotheses can be discarded.

If we apply this principle to qualitative model induction, then minimizing a model that fits the data will yield a better hypothesis according to the MDL principle.

3.1.3 Economy of Representation (Generative Linguistics)

This principle states that representations are non-redundant. The abstract representation of a surface form like "the boys walk" will only contain the feature PLURAL once, because agreement in number between the subject and verb is mandatory, and hence there is no point in storing the feature twice.

Similarly when a scenario in a qualitative model contains two trees, both should rely on one and the same dependency stating the relation between size and shade, not on two dependencies specific to their instances.

4 Approach

In this section we describe our approach on minimizing, generalizing and splitting single model fragments into modular and compositional fragments. First the problem will be stated, then an algorithm, and finally in section 5 some promising results.

4.1 Problem statement

We take a monolithic model and divide it into several smaller fragments. To do so, we need to comply to the following ranked list of constraints:[3]

1. The model should be equivalent, that is, the resulting behavior should be exactly the same as the original given the same input (a scenario).
2. The output should be as parsimonious as possible, that is, fragments should be minimal, both in size and number.
3. General fragments are preferred over specific fragments; this entails a bias for smaller fragments and modularity.

We make no claim as to the didactic and cognitive suitability of these constraints, but it is not hard to imagine that these constraints will provide an improvement when applied to any model lacking in modularity. It is well known that modularity is a sound engineering approach, and there is no reason to assume Qualitative Reasoning to be an exception. However, there are subtle choices to be made as to the exact granularity of fragments, because these constraints underdetermine the space of candidates, and because cognitive optimality would demand empirical studies about chunking and comprehension etc. We will gloss over these issues and apply the aforementioned constraints without further claims of optimality.

4.2 Terminology

instance Both quantities and entities come in classes and instances. An instance is unique, whereas a class is a generic identifier. The AM algorithm adds numeric identifiers to class names, e.g., $isa(height11, height)$ means that $height11$ is an instance of $height$.

dependency A relation between two quantities. Possible relations include influences, proportionalities, correspondences; we also include (in)equalities in this list for the sake of generality. For example: $dependency(inf\_pos, flow1, container\_right)$.

[3]Note that the terminology in this section of ranked constraints, candidates and optimality is of course a nod to Optimality Theory [14]

structural relation A structural relation is a named relation between entities. They denote the physical configuration of a system. For example a container that is connected to a pipe can be denoted as $struct\_rel(from, container_left, pipe1)$.

pivot With a pivot we intend the conditions under which dependencies hold. Whereas dependencies are the main ingredients of model fragments, the pivot provides a way to cluster these dependencies around common conditions.

4.3 Algorithm

Overview:

- Find a list of pivots, conditions on which the model fragments are based. Currently these are single structural relations.
- For each pivot, find a set of dependencies which apply whenever the pivot is present, without exception
- Generalize these dependencies into a single model fragment, several dependencies among instances may be collapsed into one

Sketch of the algorithm:

1. Input: take the set of dependencies, M, (as generated by e.g. model induction or from a monolithic model fragment created by a user).
2. Partition this set into equivalence classes according to the equivalence relation of the conditions under which each dependency holds.
3. Minimize each equivalence class by substituting $Q_1 \stackrel{I+}{\rightarrow} Q_2$ for a set such as $\{Q_1^a \stackrel{I+}{\rightarrow} Q_2^b, Q_1^c \stackrel{I+}{\rightarrow} Q_2^d, ...\}$, yielding elements of the set MFs
4. Dump the remaining dependencies in a single fragment, UF. This set should be empty if the model is properly designed, i.e., the model should not contain dependencies that apply to some instances but not to others.
5. Output: minimized model = $MFs \cup \{UF\}$

This sketch can be formalized and instantiated into a definition with set builder notation.[4] First two helper definitions. The first defines a structural relation between two entity classes:

$$einstance(R, E_1, E_2) := \exists e_1, e_2 \; [struct\_rel(R, e_1, e_2) \\ \wedge isa(e_1, E_1) \wedge isa(e_2, E_2)]$$

The second definition defines a predicate for instances of structural relations and related quantities:

[4]Note that the following definitions closely follow our Prolog implementation, as such they include certain choices and simplifications limiting the generality of the approach as previously sketched.

$$
\begin{aligned}
qinstance(R, Q_1, Q_2, q_1, q_2) := \exists e_1, e_2\ [&struct\_rel(R, e_1, e_2) \\
&\wedge has(e_1, q_1) \wedge has(e_2, q_2) \\
&\wedge isa(q_1, Q_1) \wedge isa(q_2, Q_2)]
\end{aligned}
$$

In this definition q_n refers to an instance of a quantity Q_n, similarly e_n refers to an instance of an entity E_n; such pairs are related by the $isa(a, b)$ predicate denoting that a is an instance of b. Quantities and entities are related by $has(e_n, q_m)$ denoting that e_n has the quantity q_m. Finally $struct\_rel(R, e_n, e_m)$ denotes the configuration that e_n is structurally related to e_m with relation R; for practical purposes we assume that each entity has a reflexive relation called "self," which allows the isolation of dependencies between the quantities of a single entity.[5]

The next definition defines an equivalence relation on these instances, stating that their dependencies hold universally, given their condition (which is here restricted to a single structural relation):

$$
\begin{aligned}
&samedeps(R, E_1, E_2, M) := \\
&\qquad \forall d, q_1, q_2\ [\ (dependency(d, q_1, q_2) \in M \\
&\qquad\qquad \wedge\ qinstance(R, E_1, E_2, q_1, q_2, Q_1, Q_2)) \\
&\qquad\qquad\quad \rightarrow \forall e_1', e_2'\ [\ qinstance(R, E_1, E_2, q_1', q_2', Q_1, Q_2) \\
&\qquad\qquad\qquad \rightarrow dependency(d, q_1', q_2') \in M\ ]\ ]
\end{aligned}
$$

Here $dependency(d, a, b)$ denotes a directed relation d between a and b (undirected relations can be expressed using two directed relations). Possible values for d correspond to influences, proportionalities, correspondences and equalities. While a and b usually denote quantities, note that they can also refer to an arithmetic operation between two quantities, e.g. $min(q_n, q_m)$. We gloss over this detail in these definitions.

These two predicates combined give us the set of pivots for a given monolithic model:

$$
\begin{aligned}
qpivots := \{\ \langle R, Q_1, Q_2 \rangle\ |\ & \\
&qinstance(R, Q_1, Q_2, q_1, q_2) \\
&\wedge isa(q_1, Q_1) \wedge isa(q_2, Q_2) \\
&\wedge has(E_1, Q_1) \wedge has(E_2, Q_2) \\
&\wedge samedeps(R, E_1, E_2, M)\}
\end{aligned}
$$

Instead of defining pivots on pairs of quantities, we can also use pairs of entities:

$$
\begin{aligned}
epivots := \{\ \langle R, E_1, E_2 \rangle\ |\ & \\
&instance(R, E_1, E_2) \\
&\wedge samedeps(R, E_1, E_2, M)\}
\end{aligned}
$$

[5]see the communicating vessels for an example with amount, height and pressure of a container

After defining the pivots we simply map them to their dependencies to obtain minimized, modular fragments:

$$
\begin{aligned}
qfragments := \{\ \{\ & dependency(d, Q_1, Q_2) \mid \\
& \qquad dependency(d, q_1, q_2) \in M \\
& \qquad isa(q_1, Q_1) \wedge isa(q_2, Q_2)\} \\
& \mid \langle R, Q_1, Q_2 \rangle \in qpivots \vee \langle R, Q_2, Q_1 \rangle \in qpivots\ \}
\end{aligned}
$$

Similarly for pivots of pairs of entities:

$$
\begin{aligned}
efragments := \{\ \{\ & dependency(d, Q_1, Q_2) \mid \\
& \qquad qinstance(R, E_1, E_2, q_1, q_2, Q_1, Q_2) \\
& \qquad \wedge dependency(d, q_1, q_2) \in M\} \\
& \mid \langle R, E_1, E_2 \rangle \in epivots \vee \langle R, E_2, E_1 \rangle \in epivots\ \}
\end{aligned}
$$

Note that while in this definition pivots are restricted to single triples of a relation and two quantities, the definition should be generalized so that pivots can be merged into multiple triples when they are applicable under the same conditions. This is in fact what we have done in our implementation, where we derive pivots in a bottom-up fashion for the dependencies that remain after the first pass of finding single condition pivots.

The communicating vessels model provides an example (see figure 7), where there is a dependency between two containers, even though they are not directly connected by a single structural relationship, but instead through a mutual connection to the same pipe. Our algorithm tries to find a path of structural relations between the two entities connected by the dependency in question. This is not as simple as allowing for transitive connectivity, because direction plays no role, and because cycles should be detected. The shortest path is always used as the pivot.

5 Results

The evaluation is based on a few well known example Garp models. Initially we wanted to use the output of the AM algorithm directly, but we experienced some problems with the implementations of the AM algorithm. The output of the automatic modeling implementations of [8] and [10] is not easily accessible in a simple Prolog list. The output of [8] is sent directly to a new Garp model, which makes testing easy, but the most recent implementation [10] which supports interacting processes does not return equalities and arithmetic relations among its output. The scenario and entity hierarchy is accessible through the Garp API.

To avoid having to deal with these implementation matters we have based our current proof of concept implementation on our own representation (as Prolog facts). Our implementation expects as its input a flat list of dependencies, and as background knowledge the scenario and entity hierarchy. Because of this the evaluation had to be done manually. We induced the models using the implementation of [8] so as to obtain monolithic models. An advantage to this is that we could choose the conceptually correct direction of proportionalities, which cannot be determined from the data using induction [10].

For all the models that we tested, the minimized fragments returns identical state graphs compared to the original models. This demonstrates that the minimization transformation preserves the behavior of the system.

5.1 Tree and shade growth

When the tree & shade model is minimized, the output remains a single model fragment, because there is only one entity. If however one splits on the basis of pairs of related quantities instead of on pairs of related entities, one obtains the fragments as in figure 4, given the input as in figure 3.

When a model of the full envisionment of a scenario with n trees is induced, the minimization algorithm will generalize this to the exact same fragment as for a single tree.

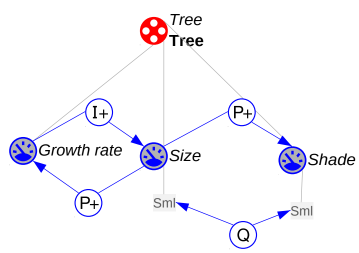

Figure 3: Monolithic model as input

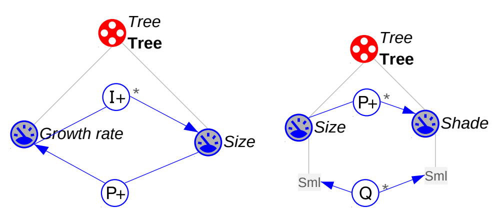

Figure 4: After minimization: two model fragments for growth and size of shade

5.2 Stacked bath tubs

For the stacked bath tub model we changed the quantity space of flow to zero-plus-max (instead of zero-plus) because this makes it possible to replace the value-correspondences by a single Q-correspondence. Value correspondences are not yet supported by the AM algorithm [10].

The algorithm correctly splits the model into two fragments, equivalent to the original model. For the input, see figure 5; for the output, see figure 6.

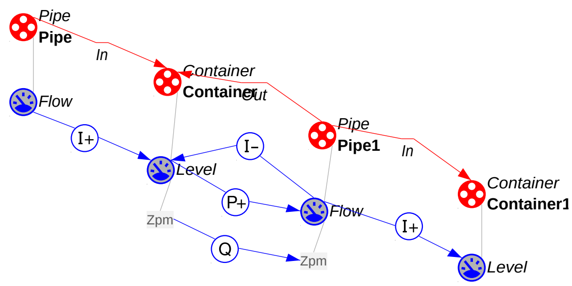

Figure 5: Monolithic bath tub model as input

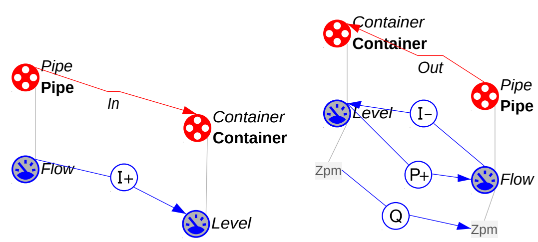

Figure 6: After minimization: two model fragments for flows going in and out respectively.

5.3 Communicating vessels

The most complicated model we have tested is the fully compositional communicating vessels model. Inducing the communicating vessels model gives almost correct output, but for some reason one equality (between height and pressure) is not found by the AM algorithm, so it has been added manually. The algorithm correctly minimizes the model into four fragments, no matter how many vessels are in the scenario. This implies that the output is truly compositional.

For the input, see figure 7; for the output minimized using quantities, see figure 8, figure 9 shows the output minimized using entities.

Three communicating vessels.

6 Future work

We will now discuss several possible extensions both to model induction in general and to model minimization in particular.

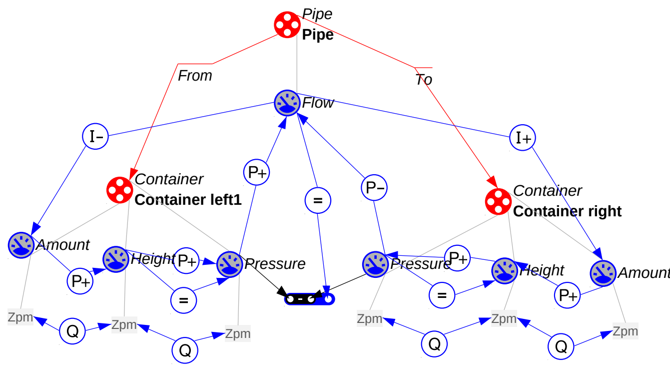

Figure 7: Monolithic model for two communicating vessels as input

6.1 Model minimization

6.1.1 Modular fragments as input

Currently the implementation expects a single monolithic model fragment as input, since that is the output as given by the automated modeling algorithm. However, this requirement could be lifted to allow partially modular fragments as input, which is especially useful when the minimization algorithm is applied as part of an interactive modeling procedure (also suggested in [9]).

This could be implemented by finding a way to merge these fragments into a monolithic fragment, while avoiding undergeneration by exploding all dependencies to specific dependencies between instances. Such an extension is fairly trivial but has not been implemented yet.

6.1.2 Inheritance: hierarchy of fragments in output

The output of our algorithm consists of a flat list of model fragments, but a more advanced method would turn this into an inheritance hierarchy, to avoid duplication and to increase re-use and clarity.

6.2 Model induction

6.2.1 Conditions

The induction algorithm has no support for conditions, except for copying structural relations between entities as found in the scenario. This severely limits the scope of models that can be induced. However, adding the induction of conditions increases the complexity greatly. This is demonstrated by an analogy to the Chomsky Hierarchy: with every addition of context-sensitivity, the complexity increases, until it reaches Turing equivalence in the unrestricted case. The Garp models that can currently be induced are the subset of possible models such that only structural relations can serve as conditions.

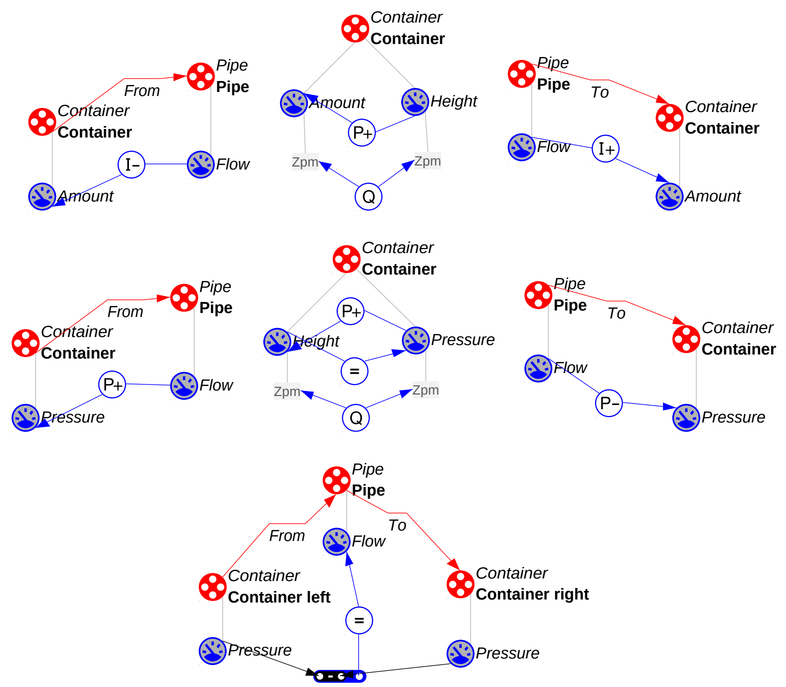

Figure 8: After minimization on quantities: seven model fragments for two communicating vessels.

On top of that, it only makes sense to add conditions to fragments, which implies that splitting into fragments (but not minimizing) would need to be integrated into the model induction algorithm itself instead of as a transformation as has been presented in the current project.

6.2.2 Interactivity

The current requirement for the induction to have access to a perfect behavior graph makes the algorithm virtually useless for practical purposes, because such a behavior graph is very difficult to specify without already having a qualitative model, which makes for a vicious circle. An alternative would be to make the induction interactive. The algorithm would formulate a series of maximally informative questions (comparable to automatic diagnosis [17]), employing the user as an informant. A series of iterations can be performed, presenting the resulting behavior graph to the user who can then highlight errors.

6.2.3 Negative exemplars

This brings us to the related problem of negative exemplars. Humans probably have an idea not only of what a system does, but also what it does not do (constraints). Negative exemplars are of course the very

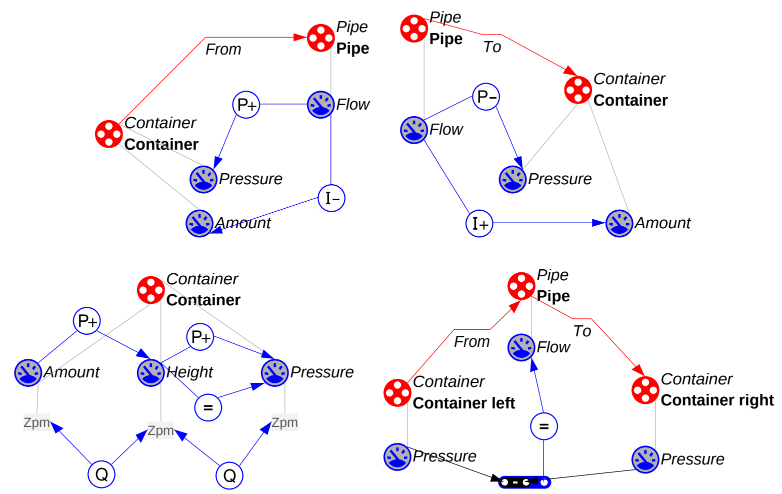

Figure 9: After minimization on entities: four model compositional fragments for communicating vessels.

reason for the problem of induction in philosophy [18, 19]. However, since qualitative models are a formalization of human knowledge, not induced from naturalistic data, negative exemplars are a resource that should be exploited. Negative exemplars could be presented by the user in the form of (in)equalities and as a list of impossible combinations of values.

6.2.4 Inductive Logic Programming

An alternative approach to model induction would be to use the general framework of Inductive Logic Programming (ILP) [11]. This framework provides a way to learn arbitrary Prolog programs based on background knowledge and exemplars of predicates. In the case of inducing a Garp model this would mean having the Garp engine as background, the behavior graph as the set of exemplars, and model fragments represented as Prolog predicates (an encoding would need to be defined) as output. In [20] (pp. 356 and on) a method of axiomatizing qualitative models is discussed, in which state transitions are first-order derivable theorems; this work could provide a good starting point for an exploration of applying ILP to automated modeling.

7 Conclusion

In this paper we have introduced an algorithm that can successfully divide complex monolithic model fragments into simple, re-usable fragments. The models on which we tested our approach were correctly splitted into fragments, which show identical behavior compared to the original models. Furthermore, the fragments correspond well with fragments made by experts.

Our approach is general enough that it applies both to Automatic Modeling and to models made by beginners. From now on, minimizing qualitative models in Garp can be done automatically, at least for the subset of models supported by the proof of concept implementation.

References

[1] Allen Newell and Herbert A. Simon, “Computer Science as Empirical Inquiry: Symbols and Search,” Communications of the ACM. vol. 19, No. 3, pp. 113-126, March, 1976. http://www.rci.rutgers.edu/~cfs/472html/AISEARCH/PSS/PSSH1.html

[2] Bredeweg, B. and Salles, P. (2009), Handbook of Ecological Modelling, Chapter 19 - Mediating conceptual knowledge using qualitative reasoning. In: Jorgen, S.V., Chon, T-S., Recknagel, F.A. (Eds.), Handbook of Ecological Modelling and Informatics. Wit Press, Southampton, UK, pp. 351.398.

[3] Bredeweg, B., Linnebank, F., Bouwer, A. and Liem, J. (2009) 02 Bredeweg EtAl ECOINF 2009.pdf (598.325 Kb) Garp3 - Workbench for Qualitative Modelling and Simulation. Ecological Informatics 4(5-6), 263-281.

[4] Forbus, K.D. (1984) Qualitative process theory. Artificial Intelligence, 24:85-168.

[5] Forbus, K.D. (1993) Qualitative process theory: twelve years after. Artificial Intelligence, 59:115-123.

[6] Kleer, J. de and Brown J.S. (1984) deKleerBrown1984.pdf (4.005 Mb) A qualitative Physics based on confluences, Artificial Intelligence, 24:7-83

[7] Cioaca, Linnebank, Bredeweg, Salles 2009 Cioaca EtAlECOINF 2009.pdf (1,001.442 Kb) A qualitative reasoning model of algal bloom in the Danube Delta Biosphere Resere (DDBR) in: *ecological informatics 4* (2009) p282-298

[8] Buisman, H., “Automated modeling in process-based qualitative reasoning” http://staff.science.uva.nl/~bredeweg/pdf/BSc/20062007/Buisman.pdf

[9] Liem, J., Buisman, H. and Bredeweg, B., “Supporting Conceptual Knowledge Capture Through Automatic Modelling”

[10] van Weelden, C., “Automated modeling of conceptual knowledge” http://staff.science.uva.nl/~bredeweg/pdf/BSc/20082009/vanWeelden.pdf

[11] Mitchell, Tom (1997), “Machine Learning,” McGraw Hill.

[12] John Hopcroft and Jeffrey Ullman (1979). “Introduction to Automata Theory, Languages, and Computation.” Addison Wesley.

[13] Russell S, Norvig P (1995). “Artificial Intelligence: A Modern Approach,” Prentice Hall Series in Artificial Intelligence. Englewood Cliffs, New Jersey

[14] Prince, Alan and Paul Smolensky (1993/2004): "Optimality Theory: Constraint Interaction in Generative Grammar." Blackwell Publishers (2004). Technical Report, Rutgers University Center for Cognitive Science and Computer Science Department, University of Colorado at Boulder (1993).

[15] Haugeland, John (1985), "Artificial Intelligence: The Very Idea," Cambridge, Mass.: MIT Press, ISBN 0-262-08153-9 .

[16] Barry Smith and Roberto Casati (1994) "Naive Physics: An Essay in Ontology." *Philosophical Psychology*, 7/2, pp. 225-244.

[17] De Kleer, J. and Williams, B. C. 1989. Diagnosis with behavioral modes. In *Proceedings of the 11th international Joint Conference on Artificial intelligence - Volume 2* (Detroit, Michigan, August 20 - 25, 1989). International Joint Conference On Artificial Intelligence. Morgan Kaufmann Publishers, San Francisco, CA, 1324-1330.

[18] Hume, David (1910) [1748]. "An Enquiry concerning Human Understanding." P.F. Collier & Son. http://18th.eserver.org/hume-enquiry.html#4. Retrieved 27 December 2007.

[19] Popper, Karl (1959). "The Logic of Scientific Discovery." pp. Ch. 1.

[20] Kuipers, Benjamin (1994). "Qualitative reasoning: modeling and simulation with incomplete knowledge," MIT Press, ISBN 026211190X, 9780262111904