Qualitative Reasoning in Tutoring Interactions

Kees de Koning and **Bert Bredeweg**
Department of Social Science Informatics (S.W.I.), University of Amsterdam
Roetersstraat 15 1018 WB Amsterdam, The Netherlands
{kees,bert}@swi.psy.uva.nl

Abstract

In various research projects, qualitative models are claimed to be beneficial for teaching systems. Thus far, little experimental research has been undertaken to investigate the usefulness of these models in actual teaching situations. In this paper, we present an experimental study that examines to what extent existing qualitative reasoning representations and techniques are sufficient for modelling the interaction between a student an a teacher when discussing the (qualitative) behaviour of physical devices. The main results are that the knowledge representations as used in qualitative reasoning are largely adequate, whereas the reasoning techniques need adaptation for teaching.

Introduction

In several projects, research is done on the use of qualitative techniques in tutoring environments (*e.g.*, QUEST (White & Frederiksen 1990), ITSIE (Sime & Leitch 1992)), as well as on the cognitive validation of qualitative reasoning techniques (*e.g.*, (Kuipers & Kassirer 1987)). However, the developments in the (separate) field of qualitative reasoning itself have largely been determined by other goals and constraints than those arising in teaching situations. To our knowledge, little research has been done on the validation of specific qualitative reasoning techniques for use in tutoring systems.

In this paper, we focus on how qualitative analysis of physics problems could be incorporated into tutoring environments. More specifically, we investigate whether the techniques currently developed in the field of qualitative reasoning are a sufficient basis for modelling the dialogue between a student and a teacher. On the basis of a qualitative reasoning model (the *norm model*) for a specific physical system, we analysed protocols of the interaction between students and teachers discussing the behaviour of this system, as obtained in an empirical study. The aim of the experiment was to find out to what extent the norm model we developed covers the knowledge involved in an actual tutoring interaction.

The Balance Domain

The specific example domain used throughout this paper is the *balance domain* (Bredeweg 1992). As an introduction to the domain, consider the balance pictured in Figure 1. On each side of the balance sits a container partially filled with water. The containers are equal in weight when empty, and have an equal outlet in the bottom. Via this outlet, the water flows out of the container, thereby decreasing the weight on that side of the balance. The flow rate of the two contained liquids can be different, according to the pressure at the bottom. As a consequence, the balance moves to different positions, but always ends up in an equilibrium. The student's task is to predict the behaviour of

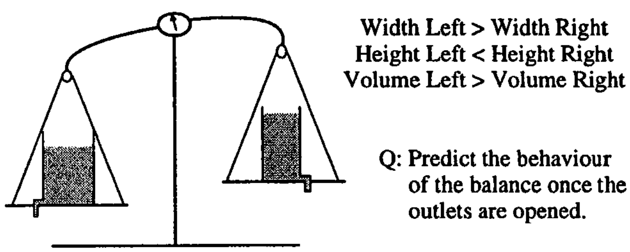

Figure 1: An Example Balance Problem

the balance, once the outlets are opened. The different

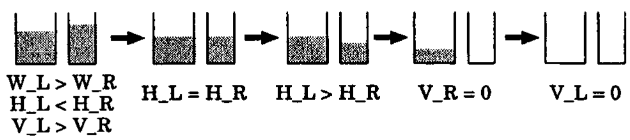

Figure 2: Different Behaviour States

qualitative states of behaviour for this balance system are defined by the behaviour of the two contained liquids, as depicted in Figure 2.

The Norm Model

The representations for qualitative behaviour analysis we used for our norm model are based on standard qualitative reasoning techniques, and implemented in a qualitative reasoning shell called GARP (Bredeweg 1992).

Representation

The knowledge representation underlying our norm model comprises the following ontological primitives. *Entities* are abstractions of the physical objects in the world. Actual "things", like a specific liquid, are defined as instances of a generic entity `liquid`. *Structural relations* can be defined between (instances of) entities, to represent for instance the fact that a liquid is contained in a container. *Features* of entities can be used to represent static properties like a container being open or closed. *Quantities* describe the behaviour of entities. For example, an entity instance `liquid` has a quantity `volume`. Quantities have qualitative values that range over ordered sets of qualitative values, called quantity spaces. Additionally, the derivative of a quantity represents the change of the quantitative value in time. *Dependencies* specify the different kinds of relations that may exist between quantities. If two quantities tend to display similar behaviour, this is modelled by a `proportionality`. A proportionality between A and B denotes that a change in A causes a change in B in the same direction, or in the case of a negative proportionality, causes a change in the opposite direction. An `influence` between A and B is used to express that the value of a quantity determines the derivative of another, indicating that if A is greater than (smaller than) zero, then B increases (decreases). A negative influence is defined likewise. A `correspondence` between two quantities A and B means that for every qualitative value of A, there is a corresponding value of B, and vice versa. In other words, when the value of A is known, B can be derived. A correspondence relation can also be directed, indicating that if A is known, B can be derived, but not vice versa. *Inequalities* are used to express actual relations between quantity values in a certain state. An inequality $A > B$ states that the (quantitative) value of A is greater than the (quantitative) value of B. Note that A and B can still have the same qualitative value.

The knowledge represented with the above primitives is organised in model fragments, constituting generic knowledge about the features and behaviour of entities from the real world in meaningful units. For instance, all knowledge concerning (the behaviour of) a liquid contained in some container is represented in the model fragment `contained liquid`. This model fragment states, among other things, that a contained liquid has a height, a width, and a volume, and that the height is proportional to the volume. Model fragments are represented as rules, where the antecedent states the conditions under which the knowledge in the consequent is applicable. For the balance domain, 13 different generic model fragments apply (*e.g.*, liquid, balance, pressure at the bottom, liquid flow).

The set of all model fragments matching a particular problem description constitutes a qualitative state. Additionally, a so-called case model (or scenario) must be provided to indicate a specific "starting point" for the system's behaviour.

Knowledge about how a qualitative state may transform into another is represented by transition rules. Three different types of rules are distinguished. *Termination rules* represent the causes for a qualitative state to end. Two different causes exist: a quantity reaches another (qualitative) value, or an inequality between two quantities changes. An instantiation of a termination rule is "because the liquid's volume is `plus` and decreasing, it will become `zero`". *Precedence (or ordering) rules* represent the order in which transitions take place. Several terminations may turn out to take place at the same moment, and hence be merged, because they are part of the same transition (for instance, the volume and height of a liquid column becoming zero). For terminations that constitute different transitions, precedence rules exist to represent their mutual order. For instance, one precedence rule states that "a quantity value going from a point to an interval always precedes one going from an interval to a point". Finally, *continuity rules* represent knowledge about how that part of the qualitative state that is not affected by a termination rule transfers to the successor state. An example of a continuity rule application is "because the liquid's volume is `plus` and decreasing, in the next state it will be `plus` and either decreasing or steady".

Reasoning

The reasoning knowledge involved in qualitative prediction of behaviour can be decomposed in two main parts, modelling and simulation.

The process of modelling employs the knowledge present in the case model to assemble the set(s) of applicable model fragments. This is done by first gathering all model fragments for which the conditions match with the case model. Second, the consequences are added to the model, and the newly added dependencies are used to calculate new values and derivatives of quantities.

Whereas the modelling phase specifies a physical system in one particular qualitatively state, *simulation* determines how the system's qualitative state changes over time. The transition rules represented above are applied. First, all termination rules that apply to the current state are gathered. All quantities and inequalities that are, by virtue of their derivative, subject to change, will fire some termination rule. This set of matching termination rules is then categorised as comprising one or more different transitions by means of the precedence rules for merging. The first transition(s) to happen are determined by means of precedence rules for ordering: pairs of transitions are compared, and the one happening later is removed. This is repeated until either only one transition remains (resulting in one successor state), or no precedence rules for ordering exist to distinguish between transitions left (resulting in multiple successor states). Finally, the set of remaining transitions is processed breadth-first. The actual transformation involves application of the termination rule(s) belonging to that transformation. Continuity rules govern the derivatives of the remaining quantities in the successor state.

The Experiment

To find out whether this norm model for qualitative behaviour analysis is adequate to support tutoring, we conducted a *Wizard of Oz* experiment as described below.

The general experimental goal was to determine the knowledge involved in the communication between student and teacher when discussing the subject matter (the behaviour of a physical device) in qualitative terms. We distinguish *terminology* and *reasoning knowledge*. The terminology is the student's labeling of his or her world view, the set of *concepts* that a student uses when dealing with behaviour prediction. The set of words (to be precise: word lemmas) as used in the communication, called the *vocabulary*, is an indication of the required terminological knowledge. The reasoning knowledge is actually all other knowledge involved in the problem solving process: behaviour prediction, like any other knowledge-intensive task, can be viewed as reasoning (*e.g.*, combination, abstraction, deduction) with the concepts of the terminology. In this research, we do not distinguish between the knowledge of the student and the teacher, nor do we discuss the influence of learning on the knowledge used: we only want to determine whether our norm model covers all the knowledge involved.

In a Wizard-of-Oz experiment, a human expert mimics the behaviour of a prospective system and communicates with the user (student) via a terminal.

Our experimental setup is visualised in Figure 3. A student and a teacher, sitting in different rooms, are

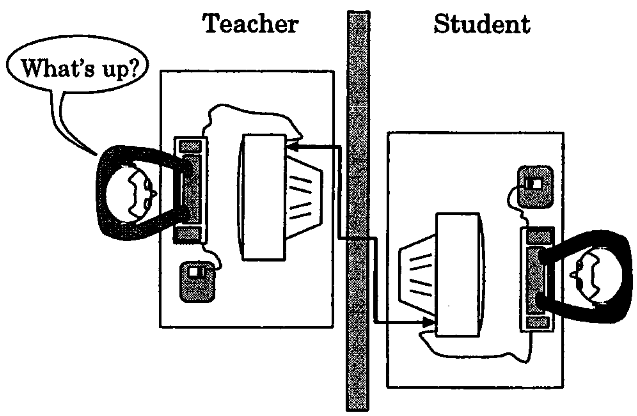

Figure 3: The Wizard of Oz Setup

communicating about the subject matter (*i.e.*, the possible behaviour(s) of the balance system) solely via a terminal. The terminal allows textual communication, as well as display of figures (in our case, pictures of different behavioural states in which the balance can be). The pictures shown on the screens are for illustration only; no graphic manipulation is possible. They are chosen from a fixed set and placed on the screen by the teacher. All communication via the terminal is saved in log files. This includes all textual communication, as well as which figures were displayed on the screen.

The experiment took one and a half hour for each student-teacher pair; there were three different teachers, and eight students. The students were first year psychology students who had learned physics in high school. They were aware of the fact that they were communicating with a human teacher; the goal of the setup used was not to fool the student in believing he or she was tutored by a machine, but in restricting the communication to a form suitable for computer-based tutoring (that is, excluding facial expressions, gestures, and intonation).

The coding is aimed at extracting the terminology and reasoning employed by students and teachers, where each reasoning step should be defined as a manipulation of terminology elements. One problem is that the reasoning steps observed in the protocols do not directly manipulate individual concepts of the terminology (like "container"), but rather manipulate *propositions* about qualitative states (like "the left container is empty"). In other words, a reasoning step can have the form "$\langle P_1 \rangle$ is the case, and therefore $\langle P_2 \rangle$", where $\langle P_1 \rangle$ and $\langle P_2 \rangle$ are propositions, and consequently not individual members of the terminology. Therefore, we need to code propositions as an intermediate step between the terminology and the reasoning.

In Figure 4, an overview is given of our *coding ontology* (the basis for the coding schemes presented in the following). The protocol sentence "The left barrel is fuller, so the volume drops faster" is used as an example. The first stage in coding is the identifica-

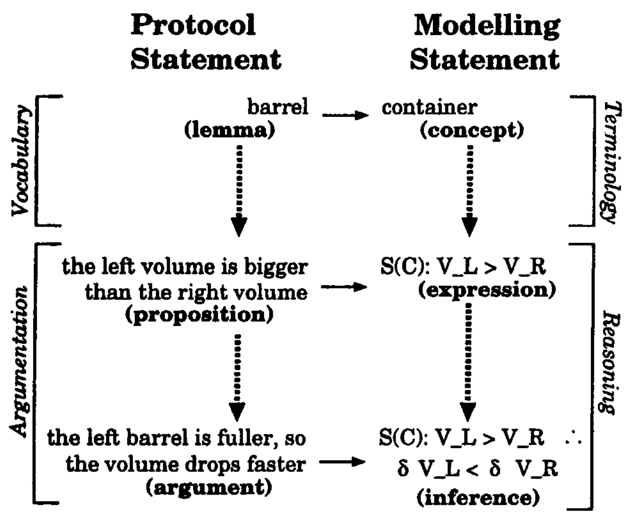

Figure 4: The Coding Ontology

tion of the **lemmas** in the protocols ("barrel"), comprising the stem word categories for all words used. A lemma maps on the modelling statement **concept** ("container"). The set of all lemmas comprises the *vocabulary*, whereas the set of all concepts comprises the *terminology*.

In the second stage, the propositions are identified. For each statement (*e.g.*, "the left barrel is fuller"), all implicit references to the context of the statement are made explicit (yielding "the left barrel is fuller than the right barrel"). To link the natural language utterances to the actual physical behaviour expression it refers to, the statements are interpreted with respect to the terminology. The interpreted **proposition** ("the left volume is greater than the right volume") is mapped onto the **expression** $S(C) : V_L > V_R$ (the syntax of expressions is explained below).

The third and final stage lays out the *argumentation* as sequences of propositions. Each **argument** ("the left barrel is fuller, so the volume drops faster") is coded as an **inference** combining two (sets of) expressions: "the left volume is greater than the right volume, therefore the (negative) derivative of the left volume is smaller than the (negative) derivative of the right volume" is coded as $S(C) : V_L > V_R \therefore \delta V_L < \delta V_R$.

Terminology Analysis

In the *lemma* count we performed on the protocols, inflections and verbal forms that are not important for the terminological interpretation were left aside.[1] This means that the words "barrel" and "barrels" are scored as equal lemmas, but also that the word "empty" will be scored as *different* lemmas in the sentences "the container is empty" (referring to the quantity space value "empty") and "the container will empty" (referring to the state termination "to empty").

In total, 41 different concepts were identified. A representative part of the results of the lemma count is given in Table 1. For each **Concept**, we depict the **Total** number of occurrences of all lemmas referring to it. We also present some typical **Examples** of the lemmas found.

Concept	Total	Examples
liquid	95	water, column
container	82	vessel, beaker
pressure	52	liquid pressure
height	46	level, column height
equilibrium	27	balance
liquid flow	3	flow
larger	159	higher, bigger
be equal	84	equally fast, the same
to depend on	23	influence, determine
now	80	this situation
state	50	initially, in the end
to change	37	to happen
to decrease	125	to flow, to drop
exercise	30	problem
picture	42	figure
therefore	171	thus, consequently
predict	10	calculate, describe

Table 1: A Concept Glossary

Starting from this set of concepts, we constructed the classification depicted in Figure 5; the emphasised terms are example lemma's. For each of the four main classes (boxed), we analysed to what extent they are covered by our norm model, and what parts of the norm model are never referred to in the protocols.

Static Subject Matter Concepts referring to one particular state of behaviour are covered completely by the norm model: each lemma that refers to the subject matter can be found literally in the norm model. Viewed from the opposite perspective, the question is whether or not the norm model is surplus to requirements. Most terminology concepts present in the norm

[1] All results presented in this paper are translated; the original protocols, as well as all analyses, are in Dutch.

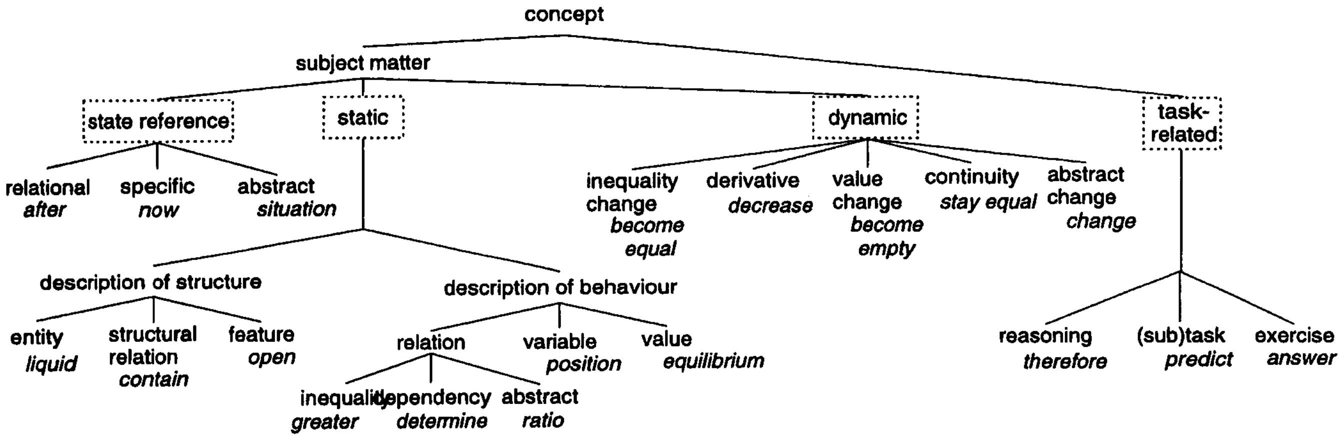

Figure 5: A Concept Type Classification

behaviour model were mentioned in the protocols. Two important aspects of the norm model representation of subject matter were not directly identifiable: *processes* and different *subtypes* of causal dependencies.

The existence of a process (in our domain either a liquid flow process, or a balance movement process) was never mentioned explicitly: there were no statements like "there is a liquid flow". However, both processes have a one-to-one relation with a specific quantity and/or derivative. Therefore, reference to the liquid flow process was always combined with statements about quantity values by statements like "water flows out of the barrel", or "the balance will move to the left".

With respect to causal dependencies, the protocols clearly show that people do not mention different types of dependencies. Nine different references were found for dependencies (*e.g.*, to cause, to determine, to influence, to depend on). "Influence" is the only lemma that could have related to one specific qualitative reasoning primitive, but from the seven references found in the protocols, it is clear that it is used by students as well as teachers to refer to all different kinds of dependencies:

Teacher: What I want to get at is that the width of the can does not influence the pressure at the bottom [...]

Hence, the differentiation in specific types of causal dependencies, as made in qualitative reasoning, is not reflected in the protocols.

While the norm model covers the complete vocabulary of the *subject matter*, there is also a set of lemmas referred to in the protocols that *do* fit into some concept class as mentioned in Figure 5, but are not within the scope of the subject matter. These *out-of-scope concepts* we found were either *entities* (*e.g.*, coil) or *quantities* (*e.g.*, friction). Out-of-scope concepts can in principle be added to the norm model. Whether this is appropriate, is a question that should be answered for each individual concept, and depends on the role it will play in a tutoring interaction.

Dynamic Subject Matter The concepts related to the dynamics of a system are also largely covered by the norm model.

Inequality changes and value changes are modelled in termination rules. Continuity of quantity values is modelled by continuity rules. Especially with respect to the latter, it is questionable whether the 'semantics' of the continuity concepts mentioned in the protocols is similar to that of the continuity rules in the norm model. For instance, when a student states that "The left barrel empties; this has no effect on the equilibrium" may be comparable to an application of a continuity rule.

Derivatives are literally present in the norm model. However, in the model, derivatives are part of the static description of behaviour, because the derivative of a quantity is simply a part of a qualitative state. In the protocols, derivatives are mentioned almost only in the context of state changes—conceptually, students view a derivative as part of the dynamics of the system, just like changes in inequality relation or changes in qualitative values. Therefore, derivatives were classified in Figure 5 under "dynamic subject matter" instead of under "static description of behaviour".

A category of concepts not covered by our norm model is that of abstract references to the system dynamics (like "to change" or "to happen"). There is no meta representation of what constitutes behaviour or change. This is part of a more general problem with our norm model, to be discussed below: the need for reflection on the prediction (process) as a whole.

State Reference Concepts referring explicitly to the notion of a qualitative state reveal a similar deficiency as the abstract references to change: although states are represented, the norm model does not facilitate reasoning *about* the set of states, or their sequence. This hampers reference to a non-specific future state, like in the proposition "Once, the levels will become equal".

The only class of state reference concepts that has a representation in the norm model, is that of state *relations*: in several (but not all) cases in which lemmas like "after" are used, this can be modelled by a *precedence rule*. For instance, the proposition "after the left one empties, the right one also empties [...]" can be seen as reference to a precedence rule.

Task-Related Concepts Concepts related to the *task* to be performed by the student can be divided in three main categories: references to the reasoning steps (therefore, because, thus), explicit mentioning of (sub)tasks (calculate, predict, describe), and concepts related to the actual exercises presented (question, answer, exercise).

How references to the reasoning steps relate to the norm model is analysed below. References to the subtasks the student should execute were rare and imprecise: one reference to the subtask "calculation", and three to the subtask "description" of the situation. In contrast, eight references were made to the main task, "prediction". None of the task-related concepts is explicitly present in our norm model. Numerous explicit references to the exercise were made. The norm model does not have a representation for what constitutes an exercise, a question, or a picture.

Reasoning Analysis

We distinguish two categories of statements, *expressions* and *inferences*. Expressions model the propositions about the domain, and are built using the terminology; inferences model the argumentation in the protocol, and are constructed by combining expressions.

A semi-formal grammar was used to extract the relevant statements from the natural language communication. The grammar, based on first-order logic, allows for representing quantity values (*e.g.*, $H_L > 0$) and derivatives (*e.g.*, $\delta H_L = -$), as well as inequality relations (*e.g.*, $V_L > V_R$) and dependency relations (*e.g.*, $P_L \propto F_L$).[2] Because we already found that no explicit differentiation is made between different dependency relations, we define only one general dependency $\propto$. Propositions are usually made in the context of one or more qualitative states. Therefore, the grammar allows indexing with states. For instance, most language statements like "The height is greater at the left" (represented in the grammar as $H_L > H_R$), are referring to the current state. In the grammar, this is represented as $S(C) : H_L > H_R$. Some examples of natural language statements and their associated expressions can be found in Table 2.

Statement	Expression
"the left barrel is fuller"	$s(C) : V_L > V_R$
"once the heights are equal"	$\exists\, s\ S(s), s \succ C : H_L = H_R$
"flow depends on pressure"	$P \propto F$
"left is empty earlier"	$\exists\, s\ S(s), s \succ C : V_L = 0 \wedge V_R = +$

Table 2: Some Examples of Expressions

Expressions

All statements in the protocols that were related to the subject matter were interpreted in terms of the semi-formal grammar. In total, 565 expressions were coded,

Expression	Example
intra-state value inequality	"the left is higher"
inter-state state termination	"the barrel becomes empty"
inter-problem entities	"it's the same balance?"
generic quantities	"the width does not matter"

Table 3: Sample Expression Categories

divided in 19 different categories. Table 3 exemplifies some expressions found in the protocols.

The organisation of expression categories is centred around what we call *primitive state expressions*: simple expressions about entities, quantities, or inequalities *within* one qualitative state of behaviour. Based on the primitive state expressions, more complex expressions can be categorised as depicted in Figure 6. These categories match the indexing terms used in Table 3. Below we discuss to what extent these four main categories are covered by the norm model.

Intra-State Expressions The expressions referring to one specific state are called intra-state expressions, and are formed by primitive state expressions indexed with one specific state (mostly the current state, $S(C)$).

[2] The quantities H, V, P, and F stand for the height, the volume, the pressure, and the flow rate of the liquid, respectively. The subscripts L and R represent the side of the balance that is referred to.

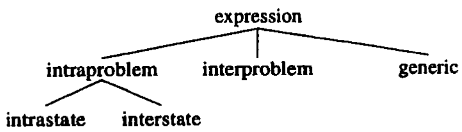

Figure 6: Higher Level Expressions

Two categories of intra-state expressions involve complex reasoning with derivatives, and are not covered by our current norm model: at the moment, there is no means for representing either an inequality between two derivatives ("left decreases faster"), or a derivative of a difference between two values ("the volume difference becomes smaller"). The other seven categories do all correspond to existing constructs in the norm model.

Inter-State Expressions Expressions that relate different states are categorised under inter-state expressions. The most common ones are those related to state transitions. The factual contents of **state terminations** ("the container becomes empty") are represented in our norm model by termination rules; that is, the fact that a container becomes empty, interpreted as "the volume of the liquid becomes zero", is derived in the norm model by means of a termination rule. However, the termination rule is part of a complete state transition, where other terminations may have been calculated, but rejected on several grounds. Often, no other information about the transformation is provided by the student. **State ordering** ("the left one becomes empty *before the right one*") and **continuity** ("the balance keeps the same position, of course") are modelled by precedence and continuity rules, respectively.

Inequalities between values in different states, like "the pressure is now lower [than in the previous state]" do not have an explicit representative in the norm model.

Inter-Problem Expressions Knowledge about previous exercises is sometimes used in the current situation. This involves statements like "This is the same as in the previous exercise", where "this" refers to one specific quantity value, the behaviour state as a whole, or even a set of behavioural states ("From here on, it is the same as before"). Because there is no explicit notion of an exercise in our current norm model, comparison of different exercises (or rather exercise solutions) is impossible.

Generic Expressions The last category of expressions are generic ones, *i.e.* expressions that do not make reference to a specific state or even a specific problem, but instead refer to the domain knowledge in general. The most important category of generic expressions involves dependencies, like in "[...], therefore equal taps do not always necessarily lose the same amount of water". Generic expressions are not easy to deal with by using the norm model of the current state: the norm model does not facilitate abstraction from the specific domain.

Inferences

Expressions are used to form the basic reasoning steps called *inferences.* In total, we found 147 inferences, divided in 63 different types. Only 6 inferences were inherently incorrect (*i.e.* logically unsound derivations), and another 25 were incorrect in their specific context (*e.g.*, the conditions were not true). The type of an inference is defined by the expression type(s) of its antecedent and its consequent. All inferences appear to be intra-problem: no explicit inferences are found about combining problems, or about generic subject matter issues. Below, the most important inference types are discussed. For an exhaustive presentation of inference types, see (de Koning & Bredeweg 1996). The syntax is given in bold; parentheses denote that the antecedent can be a set of expressions.

Inequality Correspondence
{value inequality} ∴ value inequality

By far the most frequent inference was the derivation of a value inequality from another value inequality, as in "Because the left column is higher, the pressure at the bottom is greater as well" ($S(C) : H_L > H_R \therefore P_L > P_R$). We call this kind of inference an *inequality correspondence.* Out of a total of 35, 33 of these inequality correspondences were intra-state, and hence belonging to the specification of the current state; the other two inferences in this category relate an inequality in the current state to one in the next state, hence referring to a *termination.*

In the norm model, inequality correspondences are derived, but the procedure does not use the correspondences, but explicit equality statements. This is caused by a rather complex technical problem in qualitative reasoning when comparing similar systems; for details, see (Bredeweg, de Koning, & Schut 1995).

Corresponding Influences
{value inequality} ∴ derivative inequality

Inferences of this type always implicitly involve an *influence* relating the value of a quantity to the derivative of another quantity, as in "The flow rate will be initially greater there [at the left]. So that barrel loses water faster" ($S(C) : F_L > F_R \therefore \delta V_L < \delta V_R$). The related influence states that if the flow rate is positive, the volume will decrease. Reasoning with derivatives is not covered in the norm model; therefore, the conclusion $\delta V_L < \delta V_R$ cannot be explicitly derived. Reasoning with influences is correctly covered.

State Termination
derivative inequality ∴ value inequality

This kind of derivation refers to qualitative state terminations: derivatives are used as a cause for some value inequality to *change*, as in "[...] so that height [left] will decrease faster. Therefore, the flow rate at the left becomes smaller than at the right [...]" ($S(C) : \delta H_L < \delta H_R \therefore S(N) : F_L < F_R$). These terminations are not similar to the methods used in the norm model simulations. Therefore, the conditions as well as the conclusion can be found in the output of the simulation, but the relation between both has to be checked by other mechanisms.

Future State
{value inequality} ∴ ∃ state: value inequality

An inequality is used to derive that *once*, another inequality will hold: "The columns will, because of the faster flow rate of the left column, once become equal" ($S(C) : F_L > F_R \therefore \exists S(t), t \succ C : H_L = H_R$). Because no explicit notion of (future) states exist in the norm model, this type of inference cannot be dealt with directly.

Problems with Inferences

The covering of inferences by the norm model is, at the level of one-to-one correspondences, insufficient. At closer inspection, three main reasons can be identified that cover most problems.

1. The incapacity of the norm model to deal with complex expressions involving derivatives. This is mainly a technical, and not a fundamental problem, which can be solved by extending the simulator. We do not discuss this issue in further detail here.

2. The difference in steps (step size) taken by human reasoners and simulators. This is closely related to the difference in the control over the inferences made: the simulator calculates states and state transition in a fixed, breadth-first way, whereas people used more flexible, "best-first" strategies (see also (Bredeweg & Schut 1991)). When we carefully investigate the human inference steps that do not correspond to those of the norm model simulator, correct inferences are composed of conditions and consequences that *are* part of the norm model; only the consequences are derived differently by the simulator, and often a lot of additional knowledge that is used by the simulator (and is *required* to prove the consequences) is not mentioned explicitly by human reasoners. For incorrect inferences found in the protocol, in most cases it is clear that given the additional knowledge in the norm model, the consequences could never have been derived from the conditions.

3. The lacking ability to reason *about* states and the relations between states. The simulator can only reason about the knowledge within one state, and about the transition(s) to the next state(s). A meta-level reasoner is needed to facilitate reasoning over the prediction (exercise) as a whole.

The next section proposes extensions to the norm model that account for the second and third problem.

Extending the Norm Model

To overcome the insufficient coverage of the reasoning steps, two extensions to the norm model are proposed: *partially matching* of inferences, and *reflection* on the prediction task.

Partially Matching of Inferences Because humans do not necessarily reason in the same order and with the same precision as our norm model simulator does, we need a means for partially matching the inferences found in the protocol to the *inference graph* of the simulator. The idea of a partial match is that a student's inference may be interpreted as being correct, when only a *subset* of the required antecedents are mentioned. To exemplify this, we use a graphical representation for the inference graph derived by the simulator, as is shown in Figure 7. Inference graphs

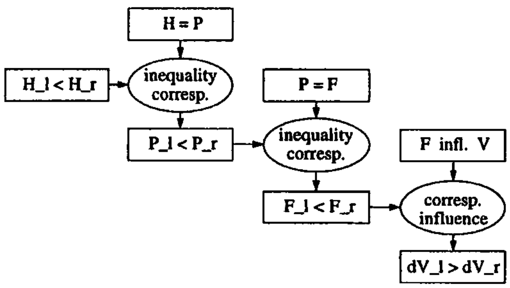

Figure 7: Inference Graph

can be generated for complete predictions, including all intra-state reasoning and state transitions. Inter-problem comparison can be done by comparing different inference structures. In the context of the problem situation given in Figure 1 and 7, the depicted chain of inferences was made as follows:

Student: The column right is higher, so the pressure is higher as well. Therefore, the flow rate will be initially higher there. So that barrel loses water the fastest [...]

Using inference graphs, we can identify which knowledge is used in deriving a certain expression, and check whether the conditions found in a student's inference are at least a subset of the conditions used in the simulator. Matching the inferences now becomes a focussed search in the inference graph to a connection between the conditions and the conclusion.

This does not yet solve all problems. An important problem is how to determine whether a student's inference covers enough parts of the knowledge actually involved in its counterpart in the inference structure. For example, the inference "The right column is higher, so the pressure is higher as well" does not explicitly mention the correspondences between height and pressure at both sides, but is nevertheless always approved of by the teacher. The inference "because the right volume is smaller, that one will be empty first" is also part of a correct inference, but here the teacher asks for clarification: the volume being smaller at the right is not enough reason to derive that it will be empty first.

Summarising, inferences are interpreted by partial matches on the inference graph. Assuming that all expressions used in the conditions and the conclusion of the inference are proved valid in the norm model, this may yield three different outcomes:

- **No match is found**, indicating that the conclusion cannot be derived from the conditions by any means, and the student's answer is rejected;
- The match is **sufficient**, and the student's answer is accepted;
- The match is **underspecified**, and the student should be asked for additional argumentation.

A method for determining whether a partial match is sufficient or underspecified may rely on *a priori* knowledge about how important different expressions are within an inference, and possibly on a *student model* that contains information about the student's knowledge with respect to the "missing" conditional knowledge in the inference. The exact implementation of this method falls outside the scope of this paper.

Reflection For the purpose of tutoring, we need a slightly different view on the prediction task. Instead of focussing on the current state, and working on transitions to the immediately following state(s) only, we should view prediction of behaviour more as incrementally specifying the state transition diagram, which can be represented as a *directed graph of states*. That is, the simulator should explicitly be aware of the notion of a prediction as consisting of a set of states, connected in a directed graph, of which at the start only one member is known: the begin state. During the problem solving process, knowledge is added about other nodes of the graph, and about the edges between them. For instance, a student can start with stating that "in the end, both containers will be empty of course, and the balance will be in its equilibrium", without having said anything about the intermediate qualitative states.[3]

The idea of using a directed graph (a state transition diagram) is not new (*e.g.*, (Harel 1987)). Also, it does not imply that we need to design a new prediction engine—it is a high level view on the process that is beneficial in the context of *tutoring*. As such, we can use existing techniques to generate the set of states (either off-line or on-line), but we need a meta level reasoner that processes a prediction of behaviour as a whole, rather than piece by piece without looking ahead or back. In the example given at the beginning of this section, the *verification* of the proposition about the end state can be done by doing a total envisionment, but the meta reasoner can decide what to do with the remark (ask for an argumentation, ask for the consequences) on the basis of its knowledge that the student knows (at least) the end state. The sole fact that the student has determined the end state does not guarantee that all intermediate states and transitions are also understood. For a simulator, the only way to determine the end state is to calculate all intermediate states first, but people can use strategies not covering all facts.

Another interesting concept is that of state ordering. Numerous remarks in the protocols refer to non-specific state ordering ("left is empty earlier than right"). In graph terminology, this remark is interpreted as "there exist two nodes S_i and S_j, where S_i refers to the state in which the left container is empty, but the right one is not, and S_j to the state in which both containers are empty. There is a directed path from S_i to S_j." Furthermore, it is known that there is a path from the current state to S_i. To clarify the examples, consider the picture of the directed graph shown in Figure 8. The specific balance problem that is presented to the student in this case is pictured in Figure 1 and 7. The "student trace" graph is the result of the following interaction:

Teacher: Can you tell me what will happen here?

[3] In continuous systems, the state transition diagram contains loops, and no end state exist. Although we always mention an end state in the examples, state transition diagrams do not require its existence.

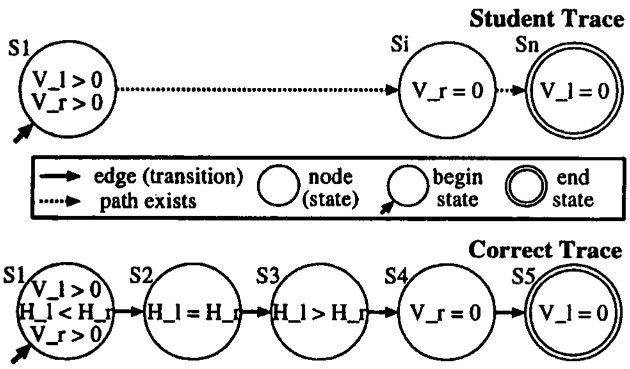

Figure 8: A Prediction Graph

Student: Well, at least they will both become empty in the end. But I think that the right one is empty first.

When compared to the correct trace shown in the lower part of Figure 8, we can see exactly what information is still lacking at the moment: the two states between the current (begin) state S_1 and the one in which the right container becomes empty (S_i), and all transitions. Note that, although the layout may suggest different, there is no evidence yet for the fact that $S_i = S_4$, nor that the end state S_n is the fifth state: the dotted arrow between S_i and S_n denotes the existence of a path, not of an edge (transition).

Partially matching of inferences and reflection both require additional reasoners on top of our norm model.

Concluding Remarks and Outlook

We have presented an empirical study on the suitability of current qualitative reasoning techniques in a tutoring environment. Using a *Wizard-of-Oz* setup, in which a student and a teacher communicate about a problem solving task via computer terminals, we have investigated whether the terminology and reasoning used in the textual interaction can be covered by a model of that problem solving task that is built by using current qualitative reasoning techniques.

At a global level, the terminology as used by students and teachers in the interaction seems to be covered reasonably well by our existing norm model. The reasoning knowledge, however, does not match very well. Especially the *control* over the reasoning process is different. Two meta level extensions should be added to the simulator to account for the problems found: a mechanism for partially matching of inferences, and a reflection component for reasoning over states.

These extensions will be implemented on top of our current norm model. The resulting model will be used to generate inference graphs of the norm behaviour. The ultimate goal is to use these inference structures as 'device models' for cognitive diagnosis (de Koning, Breuker, & Bredeweg 1995).

Acknowledgements

The ideas presented in this paper have been developed in cooperation with Joost Breuker. We would also like to acknowledge the teachers and students involved in the experiment.

References

Bredeweg, B., and Schut, C. 1991. Cognitive plausibility of a conceptual framework for modeling problem solving expertise. In *Proceedings of the 13th Conference of Cognitive Science Society*, 473–479. Hillsdale, New Jersey: Lawrence Erlbaum.

Bredeweg, B.; de Koning, K.; and Schut, C. 1995. Modelling the influence of non-changing quantities. In Wainer, J., and Carvalho, A., eds., *Advances in Artificial Intelligence.* Springer-Verlag. 131–140. SBIA '95.

Bredeweg, B. 1992. *Expertise in qualitative prediction of behaviour.* Ph.D. Dissertation, University of Amsterdam, Amsterdam, The Netherlands.

de Koning, K., and Bredeweg, B. 1996. Coaching students in reasoning about physical systems. Technical report, University of Amsterdam.

de Koning, K.; Breuker, J. A.; and Bredeweg, B. 1995. Cognitive diagnosis revisited. In Greer, J., ed., *Proceedings of the World Conference on AI and Education*, 115–122. Charlottesville, USA: AACE.

Harel, D. 1987. State charts: a visual formalism for complex systems. *Science of Computer Programming* 8:231–274.

Kuipers, B., and Kassirer, J. P. 1987. Knowledge acquisition by analysis of verbatim protocols. In Kidd, A. L., ed., *Knowledge Acquisition for Expert Systems: A Practical Handbook.* New York: Plenum Press. 45–71.

Sime, J. A., and Leitch, R. R. 1992. A learning environment based on multiple qualitative models. In Frasson, G.; Gauthier, G.; and McCalla, G. I., eds., *Proceedings of 2nd International ITS Conference*, number 608 in Lecture Notes in Computer Science, 116–123. Edinburgh, Scotland, UK: Springer-Verlag.

White, B. Y., and Frederiksen, J. R. 1990. Causal model progressions as a foundation for intelligent learning environments. *Artificial Intelligence* 42:99–157.