# A Model-Building Learning Environment with Error-based Simulation


**Tomoya Horiguchi**
Graduate School of Maritime Sciences,
Kobe University, Japan
horiguti@maritime.kobe-u.ac.jp

**Tsukasa Hirashima**
Department of Information Engineering,
Hiroshima University, Japan
tsukasa@isl.hiroshima-u.ac.jp

**Kenneth D. Forbus**
Qualitative Reasoning Group,
Northwestern University, USA
forbus@northwestern.edu



### Abstract

An important purpose of science education is to help students acquire the ability to build appropriate model of systems. For facilitating such learning activity, model-building learning environments (MBE) have been developed. Recent MBEs assist students in building conceptual models of systems and simulating their qualitative behavior. However, their feedback is insufficient when an erroneous model is built because they can’t explain how its behavior is unnatural nor how to correct it. Additionally, when the model includes inconsistent constraints, they can't create simulation itself. In this paper, we introduce a MBE which overcomes these problems using two techniques we developed: a robust simulator and semantics of constraints. The former analyzes the consistency of a model and relaxes some constraints if necessary. The latter is a systematic description of physical meanings of constraints in models and provides heuristics for explaining unnatural behaviors. A prototype was shown to be useful for learning physics in a preliminary test.


## Introduction

An important aspect of scientific expertise is the ability to build an appropriate model for a given task by identifying the structure and principle behind the phenomenon. In science education, therefore, students are often given tasks which require them to build models to predict/explain the behaviors of systems. Learning to formulate, test and revise models is crucial for understanding science and makes students active. Supporting students in articulating models and refining them through experience and reflection leads them to deeper, systematic understanding of science (Collins 1996).

For facilitating such learning activity, educational systems called model-building learning environments (MBE) have been developed (Biswas, Schwartz and Bransford 2001; Bredeweg et al. 2009; Forbus et al. 2004; isee systems 1985-2012; Jackson et al. 1996). In MBEs, students are given a set of components and build their model by combining them. A GUI is usually provided which allows students to articulate knowledge using graphical representations. Each component corresponds to a basic term of some formal language. Models made by students are translated into formal expressions of the language, then their behaviors are calculated by a simulator to give feedback about the students.

In early MBEs, mathematical expressions were used as the modeling language and numeric results were derived (Costanza and Voinov 2001). However, the abstract concepts represented by mathematical expressions are relatively inaccessible to students, such as middle school students. It is also difficult for them to interpret the results of numeric calculation. Additionally, mathematical expressions can't capture many crucial aspects of models, such as the conditions under which a model is applicable. In contrast, in recent MBEs, ontological primitives of qualitative reasoning are used as a modeling language, which makes it possible to capture conceptual aspects of models' behavior, such as causality (Bredeweg et al. 2009; Forbus et al. 2004). These environments allow young students to articulate knowledge using intuitive concepts. The usefulness of MBEs have been verified through experiments in elementary and science/engineering education.

However, the feedback given by these systems is not always sufficient when students build erroneous models. They show the behavior calculated with erroneous models, but they can't explain how it is unnatural nor how to correct the error. Additionally, when models include inconsistent constraints, these systems can't create feedback by themselves. Helping students identify and correct errors in models is necessary because it is a difficult task for them (and sometimes even for teachers).

We have proposed a framework for simulating the behavior of models, including students' erroneous ideas, and for providing appropriate feedback by judging its un-naturalness. This is called 'Error-based Simulation (EBS).' (Hirashima et al. 1998; Horiguchi and Hirashima 2006) In the EBS framework, the educational implication of a behavior is judged by identifying what correct constraint it violates (i.e., how it differs from the correct behavior). Additionally, the calculability of a model is checked by nonmonotonic reasoning. If it isn't solvable (i.e., over-constrained), the simulation (which is also called 'EBS') is created by relaxing the constraint(s) responsible for the inconsistency. Some educational systems based on this framework (called 'EBS-systems') have been developed and proved effective through laboratory and classroom experiments (Hirashima et al. 1998; Hirashima et al. 2009; Horiguchi and Hirashima 2006; Horiguchi et al. 2007).

The purpose of our current research is to develop a MBE which can create appropriate feedback to students' erroneous models based on EBS framework, and to evaluate its effectiveness. Appropriate feedback means the system can simulate the behavior of any erroneous models, and explain how it is unnatural and how to correct the error. Since EBS-systems so far are implemented in domain- and problem-specific ways, they don't have general mechanisms for handling constraints nor creating explanations. In this paper, therefore, we propose a method for creating EBSs for any erroneous models written in a modeling language. As for the language, we adopt the primitives of qualitative reasoning. Additionally, in order to explain behavior which violates some constraint, the physical meaning of each constraint needs to be explicitly described. We therefore also propose a method for describing such knowledge systematically. Finally, we report the result of a preliminary experiment to evaluate the usefulness of our method.

## Problems to be Solved

In order to simulate the behavior of an erroneous model and to explain its unnaturalness, two problems must be solved:

(1a) Erroneous models are sometimes unsolvable, that is, no solution exists that satisfy all the constraints they include. In such a case, simulation can't be created.

(2a) It is difficult to predict all possible errors students would make and to prepare the explanations for them.

The EBS framework provides the following solutions (Hirashima et al. 1998; Horiguchi and Hirashima 2006).

(1b) The calculability of a model is checked by nonmonotonic reasoning. If it isn't solvable (i.e., over-constrained), the simulation is created by relaxing some constraint(s) responsible for the inconsistency. It shows some (correct) constraints must be violated according to a student's erroneous idea.

(2b) The unnaturalness of the behavior of an erroneous model is explained by identifying what correct constraint it violates (i.e., how it differs from the correct behavior).

*Example 1: A model is incalculable because of inconsistency*

For example, figure 1 shows a qualitative model which represents the relation between a species' population and its death rate (with the graphical language of our prototype called 'Evans'). It consists of constraints such as 'C1: the population is the negative integration (I-) of the death rate,' 'C2: the population is not negative' and 'C3: the death rate is a positive constant.' This model is erroneous because the death rate is correctly not constant but in inverse proportion (P-) to the population. In fact, given zero as a initial value of the population, it has no possible next state because the population can't be negative. Usually simulators stop in such a case.

In the EBS framework, on the other hand, simulation is continued by relaxing the constraint(s). For example, by relaxing C2, a quite unnatural phenomenon occurs in which the population becomes negative. It would strongly suggest that the model is erroneous by showing that if a student's idea were correct, such impossible phenomenon would have to occur. Since what constraint is relaxed has a great influence on the unnaturalness, heuristics are necessary for choosing the constraint to be relaxed. The heuristics must be an explicit description of the physical meaning of each constraint.

*Example 2: A model is solvable but violates the correct constraint(s)*

On the other hand, figure 2 shows a qualitative model which represents the relation between the amounts of water in two containers and the flow rate of water through a pipe connecting them at their bottom. It consists of constraints such as 'C1: the pressure of water at the bottom of a container is in proportion (P+) to the amount of water in it,' 'C2: the flow rate is in proportion to the difference of the pressures' and 'C3: the amount of water in a container is the integration (I+) of the flow rate.' This model is erroneous because one of the amounts of water must be the negative integration (I-) of the flow rate while the other is the positive integration (I+) (i.e., the water is moving from one container to the other). Though this model is solvable, simulating it results in a quite unnatural phenomenon in which the total amount of water isn't conserved because the amounts of water in both containers are increasing. Usually simulators can create simulation in this case. However, they can't explain what's happening.

In the EBS framework, in such a case, an erroneous model is compared with the correct one to identify the constraint(s) included in the latter but violated in the former. In this example, the constraint 'the total amount of water is constant' is identified as being violated (this constraint isn't necessary for calculation but needs to be included redundantly in the correct model for explanation). The unnaturalness is explained based on the identification. Again, the explicit description of the physical meaning of each constraint is necessary.

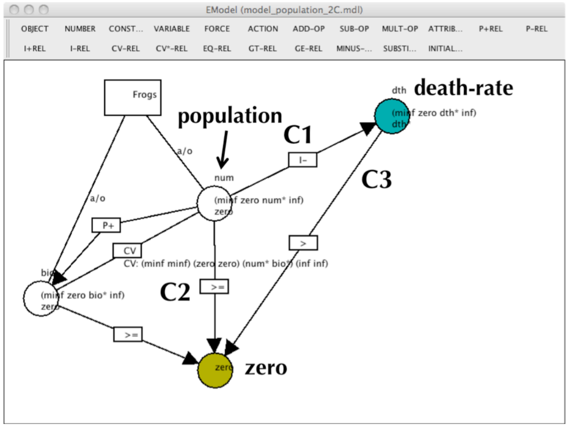


**Figure. 1.** Model of population (erroneous)

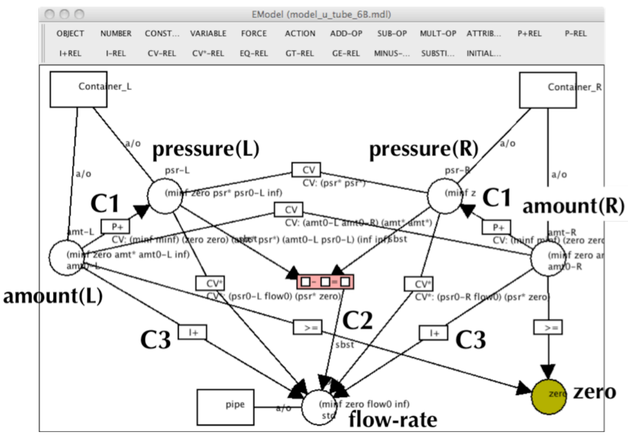


**Figure. 2.** Model of two containers (erroneous)

A simulator which has the ability in such constraint handling is called 'Robust Simulator (RSIM).' In our previous work, we developed a RSIM which can deal with the models represented with mathematical expressions (algebraic/differential equations and inequalities) by using a Truth Maintenance System (TMS) (Horiguchi and Hirashima 2006). In this research, based on the same idea, we have developed a new RSIM which can deal with the models represented with qualitative equations and inequalities. Its design and implementation are described in the next section.

In order to provide heuristics for constraint-relaxation and to explain the behavior, on the other hand, it is necessary to describe the physical meaning of each constraint included in the correct model. A guideline for such description is desirable which works in various domains. For this purpose, we developed 'Semantics of Constraints (SOC)' which is a hierarchy of 'Constraint Classes (CC).' CCs are domain-independently identified based on the roles of constraints in models. SOC is described in the section after next.

## Robust Simulator

In this research, qualitative simulation is adopted to help students understand physical phenomena qualitatively, which is important in both elementary and science/engineering education. We therefore have developed a RSIM which can deal with qualitative differential equations and inequalities (QDEs). It is called 'Qualitative RSIM (QRSIM).' QDEs are translated into a set of clauses and their consistency is analyzed by LTRE. LTRE, which is a Logic-based Truth Maintenance System (LTMS) coupled to a forward-chaining Rule Engine (Forbus and deKleer 1993), maintains the dependency network of constraints of QDEs and checks its consistency.

QRSIM derives a qualitative behavior as follows: Given a set of QDEs and initial conditions, QRSIM determines the initial qualitative state by intra-state analysis based on LTMS. Then, a set of candidates for next state is created by inter-state analysis (also based on LTMS). Such intra- and inter-state analysis are repeated alternately until a state which satisfies a terminal condition is found.

In case QDEs are erroneous, it is possible there is no candidate for the next state which is consistent. If all of the candidates are contradictory, unless no state is found which satisfies any terminal condition, QRSIM chooses one of them and relaxes its constraint to continue the simulation.

## Semantics of Constraints

A model built by a student is compared with the correct model written by an author of the teaching materials (called the 'scenario author'). The difference between them is identified as (1) extra, (2) lacking or (3) erroneous constraint(s) (that the former includes but the latter doesn't, that the latter includes but the former doesn't, that the latter includes and is erroneously written in the former, respectively). The constraint relaxed to make the model calculable is regarded as (2). As for (2) and (3), by referring to the physical meaning of the corresponding constraint in the correct model, the explanation of how the behavior of the erroneous model is unnatural is created.

Therefore, the author must not only write the correct model but also annotate the constraints it includes with their educational implication. Educational implication means the physical meaning and importance of a constraint, which are used as heuristics for both constraint-relaxation and explanation-creation. Additionally, writing the correct model means the author must enumerate all the possible constraints it is subject to. For example, in figure 2, while the constraint 'the total amount of water is constant' isn't necessary for calculation, it needs to be redundantly described for explaining the behavior. The constraints which are too obvious and often omitted also needs to be explicitly described, such as 'the amount of water in each container is not a negative number' and 'it is not greater than the capacity of the container.' This is because, based on erroneous models, any unnatural phenomenon could occur which violates such implicit constraints, such as that the value of a constant might change, and that the value of a variable might be out of its domain.

It is not easy for authors to enumerate and annotate such constraints because most of them are implicit modeling assumptions, and are usually not necessary for simulation. One possible way of helping authors is to give them a guideline for enumeration and annotation. That is, in each domain, there is a set of objects and relations which typically appear in a scenario. Each of them has a set of attributes which are typically considered in a model. Therefore, when authoring a scenario, suggesting such attributes and their default values and domains will be helpful. Additionally, there are different roles that constraints play in a model. For example, some constraints constrain the values of a set of attributes by a physical law, some constrain the domain of an attribute, and others give an initial value to an attribute. Identifying such roles helps in considering their physical meanings.

Therefore, 'Semantics of Constraints (SOC)' is necessary which is a hierarchy of 'Constraint Classes (CC).' CCs are identified based on the roles of constraints in models of physical systems. It is desirable that the SOC is independent of domains and customized for each domain. It is customized to include domain-specific relationships between constraints, such as what configuration of objects/relations/attributes in a scenario suggests what other constraints exist. Storing the examples in domains for each CC will be helpful for such customization.

We have developed the semantics by referring to the literature on qualitative reasoning (Weld and deKleer 1990), model reformulation (Falkenhainer and Forbus 1991) and engineering ontology (Mizoguchi 2002). Three major constraint classes are identified by considering the modeling process of physical systems, which are 'the constraint of physical phenomenon (PPC),' 'that of modeling assumption (MAC)' and 'that of boundary condition (BCC).' Given a physics problem (which consists of a situation and query), one assumes the structure and behavioral range of the physical system to build the model necessary and sufficient for solving the problem. Such constraints are called modeling assumptions (MAC). Some physical principles/laws which match the situation are instantiated to constrain the values of a set of attributes. These are the constraints of physical phenomenon (PPC). A set of boundary values of attributes are given to calculate the behavior of the model. These are the constraints of boundary condition (BCC).

Each of these constraint classes corresponds to a layer in the hierarchy of ontology of physical systems. As shown in figure 3, the knowledge of physical systems is often viewed as multi-layered ontology, each layer of which defines the concepts/axioms of physical systems at a particular abstraction level (Mizoguchi 2002). Boundary condition classes correspond to the most concrete level, the scenario model, which gives constraints specific to the system considered. Physical phenomenon classes correspond to the physical process ontology layer, which gives constraints specified by physical phenomena. As shown later, modeling the assumption classes correspond to various layers depending on the function of each constraint. Such correspondences with ontology layers help in considering the physical meanings of constraint classes and their violations.

In general, because a constraint in the lower layer gives more fundamental concept/axiom, constraint-violation in the lower layer could be recognized as more ‘un-natural.’ However, the scenario model is an exception. Since constraint-violation in this layer would be recognized as 'unsatisfaction of the condition given in the scenario,' which would have a great impact on students. In the following two sections, we elaborate on the constraint classes PPC and MAC to clarify their physical meanings. After that, we illustrate how the impact is estimated according to this framework when a constraint is violated.

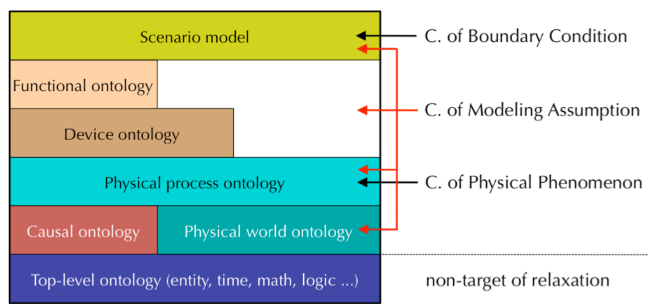


**Figure. 3.** Hierarchy of physical system ontology

## Constraints of Physical Phenomenon

### Subclasses

A physical system evolves through time, starting from an initial state. It is either changing dynamically, in a steady state or changes discontinuously. Therefore, we call the constraints in these state, 'the constraint of dynamic change (DYC),' 'that of steady state (SSC)' and 'that of discontinuous change (DCC),' respectively. Additionally, when a quantity is conserved through time, the constraint which represents that it’s amount is the same at arbitrary two time points is called 'the constraint of conservation law (CLC).'

### Violation

Because intuitions about dynamic change, steady state and conservation develop in early ages and is held steadily (diSessa 1993), their violation would be easily understood as 'unnatural phenomena.' On the other hand, as for discontinuous change, since it is hard to predict what happens in a discontinuous change intuitively, we can estimate its violation would have less impact. Therefore, it has low priority in both constraint-relaxation and explanation-creation.

## Constraints of Modeling Assumption

### Subclasses

Modeling assumptions are classified in two ways by different viewpoints: structural and functional. The former focuses on the structure of a system and its state. For example, 'the constraint of physical structure (PSC)' specifies what kind of objects, relations and attributes in a physical system are considered. On the other hand, 'constraint of operating range (ORC)' specifies the range within which the model is valid. (Examples of these constraint classes are shown in table 1.) This classification is convenient for enumerating modeling assumptions because it often suggests the descriptive aspect of a system. The latter focuses on the function of an modeling assumption. 'The constraint of process consideration (PCC)' specifies what kind of physical processes are considered in modeling. In other words, by assuming such a constraint, one ignores a physical process by putting it out of the system or into a black box, regarding its effect as a boundary condition. Additionally, 'constraint of physical world (PWC)' maintains the fundamental laws of the physical world, such as 'rigid objects never overlap,' and 'mass of an object is always greater than zero.' (Examples of these constraint classes are shown in table 2.) This classification is convenient for considering the meaning of a modeling assumptions because it suggests the mechanism of a system.

### Violation

Depending on what functional subclass a modeling assumption belongs to, the impact of its violation is systematically derived. A constraint of PCC is sometimes regarded as a given condition. In such a case, its violation is recognized as 'unsatisfaction of the given condition' (as is that of BCC). It is also regarded as a precondition of a physical process. In such a case, its violation implies the violation of PPC and is understood as 'unnatural phenomena.' On the other hand, violation of PWC is easily understood as 'impossible phenomenon.'

By using this framework, guidelines could be developed which help authors enumerate and annotate the constraints of the correct model.

**Table 1.** Constraints of modeling assumptions (structural)

| | SubClasses | SubSubClasses | Definition | Examples |
|---|---|---|---|---|
| MAC | C. of Physical Structure (**PSC**)<br><br>decision about the perspective and granularity in modeling | C. of Physical Object (**PO**) | specifies what kind of objects in a physical system are considered | -consider two blocks in contact as they are or as one<br>-consider two parallel-connected springs/resistors as they are or a compound one |
| | | C. of Physical Attribute (**PA**) | specifies what kind of relations/attributes of objects in a physical system are considered | -consider a block's net-force or its electric resistance<br>-consider the friction between two objects in contact or not |
| | C. of Operating Range (**ORC**)<br><br>decision about the behavioral range of the model | C. of Physcal Range (**PR**) | specifies the range (state space) within which the model is valid by using physical attributes | -a model of two blocks' motion where one pulls another thru a string assumes the string is taut<br>-a model of a constant resistance assumes its current & voltage are within the proportional range |
| | | C. of Conceptual Range (**CR**) | specifies the range (state space) within which the model is valid by using conceptual attributes | -a model of a block (b) descending an slope (p) by gravity from the gravitational field (g) assumes their positional relations are in(b, g), on (b, p) |

**Table 2.** Constraints of modeling assumptions (functional)

| | SubClasses | SubSubClasses | Definition | Examples |
|---|---|---|---|---|
| MAC | C. of Process-Consideration (**PCC**)<br><br>decision about what kind of physical processes are considered in modeling | C. of Process-Identification (**PI**) | specifies what kind of processes in a physical system are considered/ ignored | -Consider(rel-friction($b_1$, $p_1$)) **PA**<br>-ignore the change of form/mass of objects which collided **PO**<br>-consider the heat exchange between two objects ($T_H > T_L$) **PR** |
| | | C. of Out-Sourcing (**OS**) | ignores a physical process by putting it out of the system and regarding its effect as a boundary condition | -an outer tank which supplies water infinitely **PR**<br>-an outer power supply which always supplies 5V **PR**<br>-initial velocity $v(0) = v_0$ **PR** |
| | | C. of Black-Boxing (**BB**) | ignores a physical process by putting it in a black box and regarding its effect as a boundary condition | -consider two parallel-connected springs/resistors as they are or a compound one **PO**<br>-consider two blocks in contact moving with internal force as one **PO/CR** |
| | C. of Physical World (**PWC**)<br><br>Necessity for maintaining the fundamental law of the physical world | | maintains the fundamental law of the physical world | -rigid objects never overlap **CR**<br>-mass > 0, μ > 0, 0 ≤ e ≤ 1 **PR**<br>-g = 9.8[m/s/s], gas constant = 8.3 [J/K mol] **PR** |

## Preliminary Test

We have developed a prototype of MBE with the functions described above. Since it has a provisional GUI and no facility for scaffolding/coaching for model-building at present, we interviewed three teachers who teach basic physics at university about the usefulness of the system. Using three problems, including those of figure 1 and 2, the authors made the correct models and examples of erroneous models. Then the teachers are shown the simulations and explanations created by the system and asked if they would be useful for making students understand how the models were erroneous and should be corrected. For example, the explanation created for the model in figure 2 was: 'though the total amount of water must be constant, it unnaturally changes in the behavior of this (erroneous) model.'

The answers of the teachers were generally positive. All of them agreed this system would be useful for teaching basic concepts of physics. They said:

- Often students don't understand the physical meanings of constraints in modeling. So, for example, they change the boundary conditions in an ad hoc way (e.g., if '+' doesn't work, they try '-'). This is partly because when a

model is unsolvable, they can't receive any feedback except the fact that it's no good. In such a case, this system would help.
- Since such concepts as normal force, tension and friction aren't intuitive, it would be helpful to show what role they play in normal behaviors.
- In modeling large systems, it is helpful to examine what effect each constraint has on the whole behavior (i.e., what'll happen if it is/isn't). This system would be a convenient tool for this purpose.

The teachers also pointed out students would have much difficulty in building a model in the form of such a network even if a sophisticated GUI was given. Therefore, as indicated by the prior work (Carney 2002; Liem, Linnebank and Bredeweg 2009), providing help and scaffolding/coaching is indispensable to make this prototype of practical use. Developing a curriculum in which students can learn the components step-by-step would be a promising way (Liem, Linnebank and Bredeweg 2009).

## Concluding Remarks

In this paper, we presented a model-building learning environment (MBE) with adaptive feedback to erroneous models. It can control and explain the unnaturalness of the behavior of models by handling constraints based on their semantics. A preliminary test suggested it is a promising tool for teaching physics.

As for feedback for students' erroneous models, DynaLearn also addresses this issue (Gracia et al. 2010). In DynaLearn, a student's model is compared with the correct model built by a teacher to detect the difference between them. Missing/superfluous elements in the student's model are pointed out based on the comparison. Incorrectness of terminology and hierarchy of elements are also indicated. However, how and why the simulated behavior is unnatural cannot be explained because DynaLearn lacks the knowledge about the physical meanings of constraints in models. Additionally, if a student's model is over-constrained (i.e., includes inconsistent constraints), its behavior cannot be simulated. On the other hand, our system can not only explain the behavioral unnaturalness based on the Semantics of Constraints, but also simulate the behavior of a over-constrained model by constraint-relaxation.

Our primary future work is to refine the system and test it in classrooms (especially, the functions for assisting students in building models and presenting simulation/explanation in an easy way to understand).

## References


Biswas, G., Schwartz, D., Bransford, J. 2001. Technology Support for Complex Problem Solving - From SAD Environments to AI. In Forbus, K.D. and Feltovich, P.J. eds. *Smart Machines in Education*. 72-97. AAAI Press.

Bredeweg, B., Linnebank, F., Bouwer, A. and Liem, J. 2009. Garp3 - Workbench for qualitative modelling and simulation. *Ecological Informatics* 4(5-6): 263–281.

Carney, K. 2002. When is a Tree a Process? Influences on Student Representations of Process in "Low Floor" Qualitative Modeling Tasks. In Proceedings. of ICLS, 49-56.

Collins, A. 1996. Design Issues for Learning Environments, In Vosniadou, S., Corte, E.D., Glaser, R. and Mandl, H. eds. *International Perspectives on the Design of Technology-Supported Learning Environments*. 347-361. Lawrence Erlbaum.

Costanza, R. and Voinov, A. 2001. Modeling ecological and economic systems with stella: Part iii. *Ecological Modelling*, 143(1-2): 1–7.

diSessa, A.A. 1993. Toward an Epistemology of Physics, *Cognition and Instruction* 10(2-3): 105-225.

Falkenhainer, B. and Forbus, K.D. 1991. Compositional Modeling: Finding the Right Model for the Job. *Artificial Intelligence* 51: 95-143.

Forbus, K.D. and deKleer, J. 1993. *Building Problem Solvers*. MIT Press.

Forbus, K.D., Carney, K., Sherin, B. and Ureel, L. 2004. Qualitative modeling for middle-school students. In Proceedings of QR2004.

Gracia, J., Liem, J., Lozano, E., Corcho, O., Trna, M., Gómez-Pérez, A., and Bredeweg, B. 2010. Semantic Techniques for Enabling Knowledge Reuse in Conceptual Modelling. In Proceedings of ISWC2010, 82-97.

Hirashima, T., Horiguchi, T., Kashihara, A. and Toyoda, J. 1998. Error-Based Simulation for Error- Visualization and Its Management. *Int. J. of Artificial Intelligence in Education* 9(1-2): 17-31.

Hirashima, T., Imai, I., Horiguchi, T. and Toumoto, T. 2009. Error-Based Simulation to Promote Awareness of Errors in Elementary Mechanics and Its Evaluation. In Proceedings of AIED2009, 409-416.

Horiguchi, T. and Hirashima, T. 2006. Robust Simulator: A Method of Simulating Learners' Erroneous Equations for Making Error-based Simulation. In Proceedings of ITS2006, 655-665.

Horiguchi, T., Imai, I., Toumoto, T. and Hirashima, T. 2007. A Classroom Practice of Error-based Simulation as Counterexample to Students' Misunderstanding of Mechanics. In Proceedings of ICCE2007, 519-526.

Jackson, S.L., Stratford, S.J., Krajcik, J. and Soloway, E. 1996. A Learner-Centered Tool for Students Building Models. *Communications of the ACM* 39(4): 48-49.

Liem, J., Linnebank, F.E., and Bredeweg, B. 2009. Model Building Experiences using Garp3: Problems, Patterns and Debugging, In Proceedings of QR2009, 68-75.

Liem, J., Beek, W. and Bredeweg, B. 2010. Differentiating Qualitative Representations into Learning Spaces. In Proceedings of QR2010, 37-46.

Mizoguchi, R. 2002. Ontology-based systematization of functional knowledge. In Proceedings of TMCE2002, 45-64.

Weld, D.S. and deKleer, J. 1990. *Readings in Qualitative Reasoning about Physical Systems*. Morgan Kaufmann.