Learning and Reasoning with Qualitative Models of Physical Behavior

Scott E. Friedman, Kenneth D. Forbus, & Jason Taylor
friedman@northwestern.edu, forbus@northwestern.edu, jason.taylor@eecs.northwestern.edu
Qualitative Reasoning Group, Northwestern University
2133 Sheridan Road, Evanston, IL USA

Abstract

Building models of the physical world from examples is an important challenge for qualitative reasoning systems. We describe a system that can learn intuitive models of physical behaviors from a corpus of multimodal, multi-state stimuli, consisting of sketches and text. The system extracts and temporally encodes exemplars from the stimuli and uses analogical generalization to abstract prototypical behaviors. Using statistical analysis, the system parameterizes these abstractions into qualitative representations for reasoning. We show that the explanations the system provides for new situations are consistent with those given by naïve students.

Keywords: Cognitive modeling; conceptual change; misconceptions; naïve physics; qualitative reasoning

Introduction

Many people have intuitive models of physical domains that are at odds with scientific models (Smith, diSessa, & Roschelle, 1994; diSessa, 1993; Brown, 1994; Vosniadou, 1994). While productive for reasoning about everyday physical phenomena, these naïve models cause patterns of misconceptions. These misconceptions may result from improperly generalizing or contextualizing experience (Smith, diSessa, & Roschelle, 1994) or from incorporating instruction into a flawed intuitive framework (Vosniadou, 1994). Understanding how such intuitive models come about is an important problem for understanding conceptual change (Forbus & Gentner, 1986).

Computational models of conceptual change (e.g. Esposito et al., 2000; Ram, 1993) tend to describe how existing concepts are changed, but not how those initial concepts are learned. We believe it is important for such models to encompass the learning of the initial concepts, to reduce tailorability. This paper describes a simulation of learning intuitive physics models from experience. Experiences are provided as combinations of sketches and natural language, which are automatically processed to produce symbolic representations for learning. The encoding process is centered on the concepts to be learned, and it constructs qualitative representations of behavior across time as exemplars. Analogical generalization is used with a statistical criterion to induce abstract models of typical patterns of behavior, which constitutes our representation of intuitive models. These models can be used to make predictions and perform simple counterfactual reasoning. We compare the system's explanations to those of human students on reasoning tasks from Brown (1994) and the Force Concept Inventory (Hestenes et al., 1992).

We next briefly summarize the relevant aspects of qualitative process theory and structure-mapping theory used in the simulation. Then we describe how our stimuli are represented and encoded, motivated by results and ideas from the cognitive science literature. The learning process itself is described next, followed by how these models are used in reasoning. We show that the system's explanations of two physical situations are compatible with student explanations. We close with related and future work.

Qualitative Process Theory

People's intuitive physical knowledge appears to rely heavily on qualitative representations (Forbus & Gentner, 1986; Baillargeon, 1998). Consequently, we use qualitative process theory (Forbus, 1984) as part of our model. The learning we model here is what provides the foundation for ultimately learning physical processes; in the framework of Forbus & Gentner (1986), we are modeling the construction of *protohistories* to describe typical patterns of behavior from experience, and building on those a *causal corpus* consisting of causal relationships between those typical patterns. To represent these patterns of behavior, we use the concept of *encapsulated history* (EH) from QP theory.

An encapsulated history represents a category of abstracted behavior, over some span of time. Unlike model fragments, EHs can mention time explicitly, referring to multiple qualitative states and events. The *participants* are the entities over which an EH is instantiated. The *conditions* are statements which must hold for an instance of the EH to be *active*. When an instance of an EH is active, the statements in its *consequences* are assumed to be true. We use encapsulated histories as explanatory schemata: When instantiated, they provide an explanation for a behavior via recognizing it as an instance of a typical pattern. Furthermore, they can predict possible causes and consequences of a behavior, and hypothesize hidden conditions when a behavior is known to be active.

Since EHs can include multiple qualitative states, they can be used for learning causal relationships between behaviors and properties of the world. In naïve mechanics, for example, the models of movement, pushing, and blocking learned by the simulation are represented by EHs.

Figure 1 illustrates an EH learned by the simulation. This can be read as: *P1* pushes *P2* while *P1* and *P2* touch; the direction *dir1* from the pusher *P1* to the pushed *P2* matches the direction of the push; and pushed *P2* consequently moves (*M1*) in the direction *dir1* of the push. When given a test scenario, the system checks its learned EHs to determine whether its participants match entities in the scenario. If so, instances of those EHs are created. Each EH instance is active only if the statements in its conditions hold in the scenario. If the consequences fail to hold, that is a prediction failure of an active EH.

Encapsulated history consequences may contain typicality expressions, such as the `Normal-Usual` attribute in Figure 1. Inferring this consequence in a scenario context indicates that the phenomenon (here, the `PushingAnObject` event) has been explained by an encapsulated history.

```
define-encapsulated-history Push05
Participants:
Entity(?P1), Entity(?P2), PushingAnObject(?P3),
Direction(?dir1), Direction(?dir2)

Conditions:
providerOfMotiveForce(?P3, ?P1),
objectActedOn(?P3, ?P2),
dir-Pointing(?P3, ?dir1),
touches(?P1, ?P2),
dirBetween(?P1, ?P2, ?dir1),
dirBetween(?P2, ?P1, ?dir2)

Consequences:
Normal-Usual(and(PushingAnObject(?P3),
                 providerOfMotiveForce(?P3, ?P1),
                 objectActedOn(?P3, ?P2)))
causes-SitProp(Push05,
                (exists ?M1
                  (and MovementEvent(?M1),
                       objectMoving(?M1, ?P1),
                       motionPathway(?M1, ?dir1)))
```

Figure 1: An encapsulated history relating pushing and movement.

Analogical Generalization

Our hypothesis is that people use analogical generalization to construct encapsulated histories. To model this process, we use SEQL (Keuhne et al., 2000). SEQL is grounded in structure-mapping theory (Gentner, 1983), and uses the Structure-Mapping Engine, SME (Falkenhainer et al., 1989). Given two representations, a base and a target, SME computes a set of mappings that describe how they can be aligned (i.e. correspondences), candidate inferences that might be projected from one description to the other, and a structural evaluation score that provides a numerical measure of similarity. SEQL uses SME as follows. SEQL maintains a list of exemplars and generalizations. Given a new exemplar, it is first compared against each generalization using SME. If the score is over the *assimilation threshold*, they are combined to update the generalization. Otherwise, the new exemplar is compared with the unassimilated exemplars. Again, if the score is high enough, the two exemplars are combined to form a new generalization. Otherwise, the exemplar is added to the list of unassimilated exemplars. The combination process maintains a probability for each statement in a generalization, based on how frequently it occurred in the exemplars generalized within (Halstead & Forbus, 2005). These probabilities are used in our simulation for doing statistical tests.

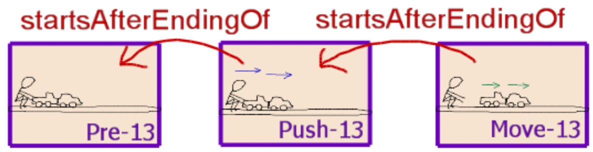

Figure 2: A sketched behavior

Multimodal Stimuli

To reduce tailorability, we provide experiences to the simulation in the form of sketches (e.g. Figure 2) accompanied by natural language text. This serves as an approximation to what learners might perceive and hear in the world. The sketches are created in CogSketch[1] (Forbus et al., 2008), an open-domain sketch understanding system. In CogSketch, users draw and label *glyphs*, objects in the sketch, to link the content of the sketches to concepts in CogSketch's knowledge base[2]. CogSketch automatically computes qualitative spatial relations between the glyphs such as topological relations (e.g. touching), relative size, and positional relationships (e.g. above).

Sketched behaviors are segmented into distinct states according to qualitative differences in behavior (e.g. changes in contact and actions of agents) to accord with findings in psychological event segmentation (Zacks, Tversky, & Iyer, 2001). Each state is drawn as a separate sketch. The sequential relationships between them are drawn as arrows on the *metalayer*, where other sub-sketches are treated as glyphs, as Figure 2 illustrates. The child, truck, and car are glyphs in the sketched states. The two right-pointing arrows in state *Push-13* are *pushing* annotations, and the two right-pointing arrows in state *Move-13* are *velocity* annotations.

Two lines of evidence motivate our encoding of the physical phenomena of pushing, movement, and blocking as separate concepts. diSessa (1993) notes that people are unlikely to confuse successful resistance (i.e. a wall blocking a person's push) from nonsuccess (i.e. a ball moving due to tugging a string) in recalling events, and that these phenomena are encoded separately. Talmy (1988) attributes this separation of success and nonsuccess encoding to varying language schemata between the two conditions.

For information not easily communicated via sketching, we use simplified English, which is converted to predicate calculus via a natural language understanding system (Tomai & Forbus, 2009). One sentence used in conjunction with the sketch in Figure 2 is, "The child child-13 is playing with the truck truck-13." The special names **`child-13`** and **`truck-13`** are the internal tokens used in the sketch for the child and the truck respectively, so that linguistically

---

[1] CogSketch is available online at http://spatiallearning.org/projects/cogsketch_index.html

[2] CogSketch uses a combination of knowledge extracted from OpenCyc (www.opencyc.org) and our own extensions for qualitative, analogical, and spatial reasoning.

expressed information is linked with information expressed via the sketch. This sentence leads to these assertions being added to the exemplar:

```
(isa truck-13 Truck)
(isa play1733 RecreationalActivity)
(performedBy play1733 child-13)
(with-UnderspecifiedAgent play1733 truck-13)
```

If the NLU system finds an ambiguity it cannot handle, it displays alternate interpretations for the experimenter to choose. No hand-coded predicate calculus statements are included in the stimuli.

This method of simulation input has limitations: Sketches are less visually rich than images, and they do not provide opportunities for the learner to autonomously experiment. Nevertheless, we believe that this is a significant advance over the hand-coded stimuli typically used by other systems, given the reduction in tailorability. These multimodal stimuli are used by our system as examples for learning and as scenarios for reasoning.

Learning

The system is provided with a set of target phenomena to learn, here *pushing*, *movement*, and *blocking*. We assume that for a truly novice learner, words used in contexts of behaviors that they do not understand are clues that there is something worth modeling.

Given a new stimulus, the system finds all instances of target phenomena that it describes, and generates an exemplar for each instance. Since an instance of a particular phenomenon may continue across state boundaries, these occurrences can span multiple states. Temporal relationships between these occurrences are derived to support learning of preconditions and consequences. For example, consider a series of states S_1-S_3, where a man is pushing a crate in S_1-S_2 and not in S_3, and the crate moves in S_2-S_3 but not in S_1. The motion would have a `startsDuring` relationship with the pushing. Each stimulus observed by the simulation is automatically temporally encoded into exemplars using this strategy.

Generalizing behaviors

For each target phenomenon, the system maintains a separate instance of SEQL, a *generalization context* (Friedman & Forbus, 2008). A generalization context has an entry pattern that is used to determine when an exemplar is relevant. For example, the entry pattern for *pushing* is:

```
(and (isa ?x PushingAnObject)
     (providerOfMotiveForce ?x ?y)
     (objectActedOn ?x ?z))
```

Figure 3 shows the generalization contexts and their contents after the learning experiment described below. Our system currently operates in batch mode, not attempting to construct models until after all of the stimuli have been processed.

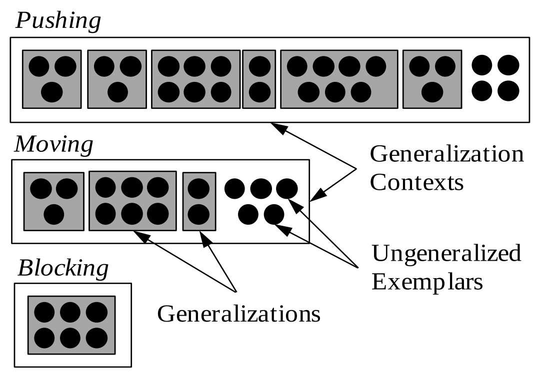

Figure 3: Generalization contexts after learning

Constructing intuitive models

The system creates encapsulated histories from generalizations in two steps: (1) Statistics are used to determine which generalizations are worth modeling with EHs, and (2) worthwhile generalizations are parameterized to create EHs. We discuss each step in turn.

Filtering generalizations

Not all SEQL generalizations can be parameterized into useful encapsulated histories. Some generalizations are overly broad, and would result in EHs that make inaccurate predictions. Consequently, the system filters out overly broad generalizations using the probability information constructed during generalization.

Generalizations are filtered by identifying correlated phenomena within generalizations and measuring the phenomena's correlation across generalizations. We assume a probability threshold t (here, 0.9) for correlation. That is, if any target phenomenon p is in a generalization with probability $P(p) \geq t$, then p is considered a *correlated phenomenon* within that generalization's context. A generalization is *decisive* if the binary entropy of all correlated phenomena p are less than the binary entropy of t, or $H(P(p)) \leq H(t)$. Entropy is the appropriate criterion to use because it measures information gain (i.e., low entropy implies high gain). Only decisive generalizations are parameterized into encapsulated histories.

Extracting Causal Models from Generalizations

The system creates one encapsulated history per decisive generalization. Expressions whose probability is lower than the probability threshold t (here, 0.9) are excluded from the EH, thus reducing contingent phenomena. Expressions that remain are analyzed to determine what role they should play in the encapsulated history.

An expression is held to be either (a) a *cause* of the state, (b) a *consequence* of the state, or (c) a *condition* that holds during the state, based on analyzing the temporal relationships involved. If an expression begins with the current state, ends with the start of the current state, or ends during the current state, it is a possible cause. If it temporally subsumes or coincides with the state, it is a possible condition. Otherwise, if it begins at any point during or immediately following the current state, it is a possible consequence.

Probabilities and temporal relationships are used to hypothesize causality. For instance, in one generalization, movement starts *with* a pushing event with P = 0.5, and starts *after* a pushing event with P = 0.5. In this case, movement is not a likely condition for pushing because it only satisfies the temporal requirement half the time, P(*starts-with*) < *t*. Conversely, movement is a likely consequence, because starting *with* and starting *after* are both permissible temporal relations of consequences, and P(*starting-with*) + P(*starting-after*) > *t*.

After the causes, conditions, and consequences are determined, the system defines an encapsulated history by introducing variables for entities that appear in the conditions, creating existence statements for the entities that appear only in the consequences, and using the generalization's attribute information to construct the participants information. Figure 1 and Figure 5 illustrate. Notice that, while the learning process removes most irrelevancies, in **`Block00`** the entity **`?P1`** is included even though it is not causally relevant. It is there because the examples involving pushing all involve the pushing agent standing or sitting on a surface – so to the system, blocking must involve touching something else.

Reasoning with Encapsulated Histories

Given a new scenario, the system attempts to make sense of it by instantiating its encapsulated histories. For each EH, it finds instances within the scenario. When an instance's conditions hold, it is active, and the statements in its **`Consequences`** are assumed to hold. This can include predicting new phenomena, as illustrated by the movement *M1* consequence in Figure 1. When constraints are violated, or consequences are not satisfied, the EH instance can be used to generate counterfactual explanations, as explained below.

To illustrate, consider a scenario used by Brown (1994) and others, illustrated in Figure 4. The sketch shows a book on a table. Gravity pushes down on the book and the table.

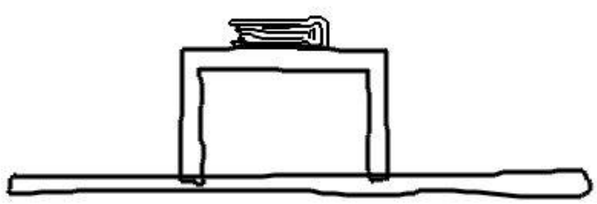

Figure 4: An example from Brown (1994) for testing learned knowledge

The scenario description includes two occurrences of pushing: gravity pushing the book and gravity pushing the table. The encapsulated history in Figure 5 can be instantiated sufficiently to be considered for inference by the simulation, since the criterion is that all non-event participants be identifiable in the scenario. Some event participants, such as pushing and blocking, need not be identified because these can be instantiated as predictions.

```
define-encapsulated-history Block00
Participants:
Entity(?P1), Entity(?P2), Entity(?P3), Entity(?P4),
PushingAnObject(?P5), PushingAnObject(?P6),
Blocking(?P7)

Conditions:
providerOfMotiveForce(?P5, ?P2),
objectActedOn(?P5, ?P3),
dir-Pointing(?P5, ?dir1),
providerOfMotiveForce(?P6, ?P3),
objectActedOn(?P6, ?P4),
dir-Pointing(?P6, ?dir1),
doneBy(?P7, ?P4),
objectActedOn(?P7, ?P3),
dirBetween(?P2, ?P3, ?dir1),
dirBetween(?P3, ?P4, ?dir1),
dirBetween(?P3, ?P2, ?dir2),
dirBetween(?P4, ?P3, ?dir2),
touches(?P2, ?P3),
touches(?P3, ?P4),
touches(?P2, ?P1)

Consequences:
Normal-Usual(and(PushingAnObject(?P5),
                 providerOfMotiveForce(?P5, ?P2),
                 objectActedOn(?P5, ?P3)))
Normal-Usual(and(PushingAnObject(?P6),
                 providerOfMotiveForce(?P6, ?P3),
                 objectActedOn(?P6, ?P4)))
Normal-Usual(and(Blocking(?P7), doneBy(?P7, ?P4),
                 objectActedOn(?P7, ?P3)))
```

Figure 5: An encapsulated history relating pushing and blocking phenomena

Specifically, activating **`Block00`** to explain gravity pushing the book requires assuming two additional events, per the conditions in Figure 5: (1) gravity ?*P2* pushes the book ?*P3* in the direction *?dir1* of the initial push, and (2) an entity ?*P4* blocks the book ?*P3*. The table alone satisfies the constraints on *?P4*, binding the last of the non-event participants. This is sufficient grounds for the simulation to instantiate new pushing and blocking events, binding them to *?P6* and *?P7*, respectively.

The simulation has two strategies for answering questions about a scenario. If the question concerns a phenomenon that is predicted by the EH instances it has created for the scenario, it answers based on that information, including any causal argument provided as part of the EH. If the question concerns some phenomenon that is not predicted, it assumes that phenomenon occurs and looks at what new EHs could be instantiated to explain it. The instantiation failures for those EH instances are provided as the reasons for the phenomenon not occurring, as shown below.

Experiment

To test whether this model can learn psychologically plausible encapsulated histories from multimodal stimuli, we observe the explanations it provides for a question from Brown's (1994) assessment of student mental models and a question from Hestenes et al.'s (1992) Force Concept Inventory. We start by summarizing human results, then describe the conditions used for the simulation, and compare the human and simulation results.

Brown's results

A question about the scenario in Figure 5 was asked of high school students: *Does the table exert a force against the book?*

Brown reported that 33 of 73 students agreed that it must, in order to counteract the downward force of the book. This is the physically correct answer. However, the 40-student majority denied that the table exerted a force. Their reasons fell into five categories:

1. Gravity pushes the book flat, and the book exerts a force on the table. The table merely supports the book (19 students)
2. The table requires energy to push (7 students)
3. The table is not pushing or pulling (5 students)
4. The table is just blocking the book (4 students)
5. The book would move up if the table exerted a force (4 students)

We query our simulation similarly, to determine whether it can reproduce some of the reasons that students gave.

Force Concept Inventory

The Force Concept Inventory (FCI) (Hestenes et al., 1992) is an assessment designed to identify student misconceptions about force. Many FCI questions involve the relationships between force, mass, and velocity, and the composition of forces to determine direction of motion. Figure 6 illustrates our sketch of question 6 from the FCI. The scenario describes a puck on a frictionless surface, moving with constant velocity, until it receives an instantaneous kick. The student must decide along which of the five paths (labeled choice-27-a/b/c/d/e below) the puck will move after receiving the kick.

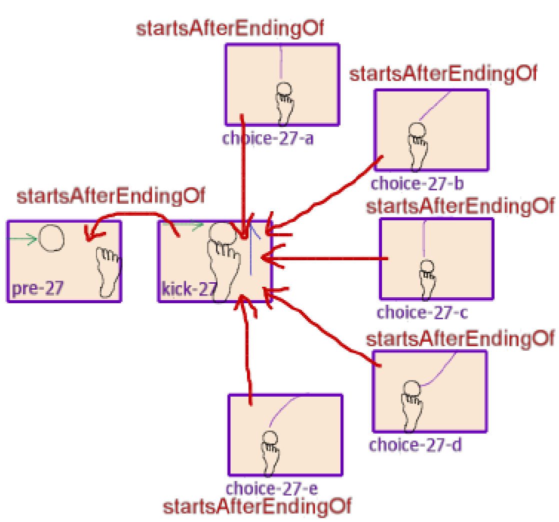

Figure 6: An example from the Force Concept Inventory (Hestenes et al, 1992)

Five pre-physics-instructed student populations, ranging from high school to college, predicted the puck would, on average:

(a) 34% - move upward, in the direction of the kick.
(b) 38% - per Newtonian principles, move diagonally.
(c) 3% - move upward and then curve to the right.
(d) 6% - gradually curve in the direction of the kick.
(e) 18% - curve in the direction of initial motion.

Other FCI questions concerned the relationships between velocity, mass, and acceleration, which were not target concepts of our simulation.

Simulation setup

Our simulation was implemented using the Companion Cognitive Systems architecture (Forbus et al., 2008), using semi-independent asynchronous agents. The Session Reasoner (the Companions agent responsible for domain reasoning) begins with 17 sketches with accompanied natural language as learning stimuli. Like Figure 2, all stimuli include pushing phenomena, and either movement or blocking phenomena. The learning stimuli did not include the test scenarios.

For each stimulus, the Session Reasoner first encodes it into exemplars, resulting in a total of 28 pushing exemplars, 16 moving exemplars, and 6 blocking exemplars. Before encoding the next stimulus, the Session Reasoner contacts the *Analogical Tickler* agent to generalize the exemplars using SEQL. The SEQL assimilation threshold was set to 0.5, which results in ten generalizations across the three generalization contexts, as illustrated in Figure 3.

After all of the learning stimuli are encoded and the exemplars are generalized, the Session Reasoner generates EHs from the resulting SEQL generalizations. The EH probability threshold was set to 0.9. Consequently, six of the generalizations were decisive, leading to the push→move model of Figure 1, the push→block model in Figure 5, and four additional models.

The four additional models learned by the system were not activated during problem solving. Three EHs describe movement behaviors caused by pushing, with minor variations in the conditions. The fourth EH describes classic "billiard ball" causality, with a push causing motion, which then causes another push and setting another entity into motion.

Both problem solving scenarios are conducted by the Session Reasoner, which tries to activate its learned EHs within the scenario contexts.

Comparison with human results

Given these EHs, how does the system perform? When given Brown's (1994) test scenario, the system activates EHs to infer the additional events of the book pushing down against the table and the table pushing down against the ground.

For Brown's query, since the simulation does not have the event of the table pushing upward against the book as a current prediction, it uses the counterfactual strategy. Only the EH of Figure 1 can provide a possible explanation. Assuming this EH is active, the simulation gets a new prediction: The book should move upward as a result of the table's push. This prediction contradicts the book's lack of motion in the scenario. Consequently, it answers that the table does not push up on the book. This is essentially the same as answer 5, given by four students.

After the proof by contradiction, the system identifies activated EHs in which the book and table jointly participate to explain their behavior in the scenario. Consequently, it uses the EH in Figure 5 to explain that gravity pushes down on the book, that the book pushes down on the table, and that the table blocks the book. This is similar to answer 4, given by four students. This explanation also resembles answer 1, given by 19 students, though the students cite the concept of support, which was not among the simulation's target phenomena. Could the system learn models corresponding to the other explanations for this scenario? If the target phenomena and corpus included the concept of support and energy, it seems likely to us that it could, but this is an empirical question. With a different corpus of examples – perhaps including examples like those used by Camp & Clement (1994) and the rest of Brown (1994) – the simulation may be capable of coming to the correct model. Answer 3 may rest on an interpretation of events being mutually exclusive, i.e., if the table is blocking, then it cannot be doing the other actions. Further experiments should clarify this.

When given the FCI scenario, the system activates the EH from Figure 1 within the "kick" state and predicts that the puck will translate in the direction of the kick during or immediately after the kick. Upon evaluating all possible following states, the system concludes that *choice-27-a* is the only successor state that fulfills this prediction. The system predicts this path for the puck, as do 34% of the FCI-assessed students in Hestenes et al. (1992), which represents the most popular misconception. The results from both scenarios support the hypothesis that the models learned by the system are like those used by physics-naïve students.

Related Work

The closest simulations are the COBWEB (Fisher, 1987) model of conceptual clustering and INTHELEX (Esposito et al., 2000), which develops and revises prolog-style theories. COBWEB does unsupervised learning of hierarchical relationships between concepts, in contrast with our use of supervised learning (via entry patterns in generalization contexts) of causal models. COBWEB calculated probabilities of features, whereas SEQL provides probabilities of structured relations. INTHELEX uses refinement operators to model multiple steps in a trajectory of learned models, whereas we focus only on one transition, the first. Both COBWEB and INTHELEX used hand-represented input stimuli, whereas ours is derived by the simulation from sketches and natural language. Ram (1993) discusses SINS, a robot navigation system that retrieves cases, adapts control parameters, and learns new associations incrementally. Both our system and SINS develop concepts incrementally from experience; however, our system learns models of physical behaviors and causal laws, while SINS learns associations between environmental conditions and control parameters.

Lockwood et al. (2005) used CogSketch and SEQL to model the learning of spatial prepositions, using single sketches labeled with words, in contrast to the sequences of sketches labeled with sentences used here.

Discussion & Future Work

We have described how analogical generalization and qualitative modeling can be used to simulate the process of learning initial intuitive models. To reduce tailorability, the simulation inputs were combinations of sketches and simplified English. The resulting answers match a subset of those of given by human students on the same scenarios.

While we believe that this is a significant first step, there is much more to be done. Other domains and physical phenomena must be incorporated, to provide more evidence as to generality. Second, we need to conduct statistical tests to determine how order-sensitive the simulation is, and how the quality of models learned varies with the number of examples provided. Additionally, modeling the induction of physical process models from the encapsulated histories learned by the system is an important step in learning intuitive physics (Forbus & Gentner, 1986).

Finally, we plan to incorporate these ideas in a larger-scale learning model, where the quality and content of its predictions guide future learning. The Companion Cognitive Systems architecture is an ideal platform for this endeavor because one of its primary goals is ubiquitous learning over an extended lifetime. With our learning and reasoning methodologies integrated into Companion Cognitive Systems, agents can use multimodal stimuli to learn new models and evaluate the productivity of existing models. These are important characteristics of a larger model of conceptual change.

Acknowledgments

This work was funded by the Office of Naval Research under grant N00014-08-1-0040. We wish to thank Dedre Gentner for discussions of concept formation with SEQL.

References

Baillargeon, R. (1998). A model of physical reasoning in infancy. *Advances in infancy research*, 3, 305-371.

Brown, D. (1994). Facilitating conceptual change using analogies and explanatory models. *International Journal of Science Education*, 16(2), 201-214.

Camp, C. W., & Clement, J. J. (1994). *Preconceptions in mechanics. Lessons dealing with students' conceptual difficulties.* Dubuque, IA: Kendall/Hunt Publishing Company.

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

Esposito, F., Semeraro, G., Fanizzi, N., & Ferilli., S. (2000). Conceptual Change in Learning Naive Physics: The Computational Model as a Theory Revision Process. In E. Lamma and P. Mello (Eds.), AI*IA99: Advances in Artificial Intelligence, Lecture Notes in Artificial Intelligence 1792, 214-225, Springer: Berlin.

Falkenhainer, B., Forbus, K. & Gentner, D. (1989). The Structure Mapping Engine: Algorithm and examples. *Artificial Intelligence*, 41, 1-63.

Fisher, D. H. (1987). Knowledge acquisition via incremental concept clustering. *Machine Learning*, 2,139-172.

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

Forbus, K. & Gentner, D. (1986). Learning Physical Domains: Towards a Theoretical Framework. In Michalski, R., Carbonell, J. and Mitchell, T. (Eds.), *Machine Learning: An Artificial Intelligence Approach.*

Forbus, K., Usher, J., Lovett, A., Lockwood, K., & Wetzel, J. (2008). CogSketch: Open-domain sketch understanding for cognitive science research and for education. *Proceedings of the Fifth Eurographics Workshop on Sketch-Based Interfaces and Modeling*. Annecy, France.

Forbus, K., Klenk, M. and Hinrichs, T. (2008). Companion Cognitive Systems: Design Goals and Some Lessons Learned. *AAAI Fall Symposium on Naturally Inspired Artificial Intelligence.*

Friedman, S. & Forbus, K. (2008). Learning Causal Models via Progressive Alignment & Qualitative Modeling: A Simulation. *Proceedings of CogSci 2008.*

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2).

Halstead, D. and Forbus, K. (2005). Transforming between Propositions and Features: Bridging the Gap. *Proceedings of AAAI-2005*. Pittsburgh, PA.

Hestenes, D., Wells, M., Swackhamer, G. (1992). Force Concept Inventory. *The Physics Teacher*, 30, 141-158.

Kuehne, S., Forbus, K., Gentner, D. and Quinn, B. (2000). SEQL: Category learning as progressive abstraction using structure mapping. *Proceedings of CogSci 2000*.

Lockwood, K., Forbus, K., & Usher, J. (2005). SpaceCase: A model of spatial preposition use. *Proceedings of the 27th Annual Conference of the Cognitive Science Society*.

Ram, A. (1993). Creative conceptual change. *Proceedings of CogSci 1993.*

Smith, J., diSessa, A., & Roschelle, J. (1994). Misconceptions Reconceived: A Constructivist Analysis of Knowledge in Transition. *Journal of the Learning Sciences*, 3(2), 115-163.

Talmy, L. (1988). Force dynamics in language and cognition. *Cognitive Science*, 12(1), 49-100.

Tomai, E. and Forbus, K. (2009). EA NLU: Practical Language Understanding for Cognitive Modeling. *Proceedings of the 22nd International Florida Artificial Intelligence Research Society Conference*.

Vosniadou, S. (1994). Capturing and modeling the process of conceptual change. *Learning and Instruction*, 4, 45-69.

Zacks, J., Tversky, B., & Iyer, G. (2001). Perceiving, remembering, and communicating structure in events. *Journal of Experimental Psychology. General*. 130(1), 29-58.