LEARNING NAÏVE PHYSICS MODELS BY ANALOGICAL GENERALIZATION

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

ABSTRACT

How do people learn intuitive models of the world from experience? We describe a simulation that uses analogical generalization to learn naïve models of pushing and blocking from experience. Experiences are represented by a type of comic strip, consisting of sequences of sketches and simplified English that are automatically encoded by the simulation. We show that the models it learns are compatible with naïve models found in the literature, and analyze the effects of presentation order.

INTRODUCTION

People learn intuitive models of physical domains through their observations and experiences in the world. Many of these intuitive models are at odds with scientific models (Smith, diSessa, & Roschelle, 1994; diSessa, 1993; Brown, 1994). While productive for explaining and predicting physical phenomena, intuitive models can cause patterns of misconceptions. These misconceptions may result from improperly generalizing or contextualizing experience (Smith, diSessa, & Roschelle, 1994). Understanding how such misconceptions are learned is an important problem for models of conceptual change and learning physical domains (Forbus & Gentner, 1986).

This paper describes a simulation of learning intuitive physics models from experience. Our hypothesis is that learners can form mental models of physical domains via analogical generalization, and that some of these models are scientifically incorrect. Our system learns in three steps: exemplar encoding, analogical generalization, and model formalization. Experiences are provided as sequences of sketches accompanied by natural language, which are automatically encoded to produce exemplars by identifying instances of behavior across time. Exemplars are generalized analogically to produce prototypical behaviors. These *protohistories* are automatically formalized into parameterized qualitative models, which 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 also analyze the effects of presentation order on the simulation.

We first briefly summarize the relevant aspects of qualitative process theory and structure-mapping theory used here. Then we describe how the stimuli are represented and encoded. The learning process itself is described next, followed by how the learned models are used in reasoning. We show that the system's explanations of two physical situations are compatible with student explanations, and we analyze the effects of changing stimulus order. 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) in our model. 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. The *participants* are the entities over which an EH is instantiated. The *conditions* are statements which must hold for an EH instance to be *active.* When an EH instance 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. In our simulation, the models of movement, pushing, and blocking learned by the simulation are represented by EHs.

Figure 1 shows 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 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.

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, ?P2),
              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 based on 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 exemplars are combined to form a new generalization. Otherwise, the exemplar is added to the list of unassimilated exemplars. A probability is maintained for each statement in a generalization, based on how frequently it occurred in the exemplars merged into it (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 a sequence 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) based on psychological findings in event segmentation (Zacks, Tversky, & Iyer, 2001). Each state is drawn as a separate sub-sketch. Sequential relationships between them are drawn as arrows on the *metalayer*, where sub-sketches are treated as glyphs (Figure 2). The child, truck, and car are glyphs in the sketched states. The two rightward arrows in state *Push-13* are *pushing* annotations, and the two rightward arrows in state *Move-13* are *velocity* annotations.

Two lines of evidence motivate our encoding 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 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

[1] CogSketch is available online at http://www.spatiallearning.org/projects/cogsketch_index.html
[2] CogSketch uses knowledge from OpenCyc (www.opencyc.org) plus our extensions for qualitative, analogical, and spatial reasoning.

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 sketch of states S_1-S_3 (similar to Figure 2) 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 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. Entry patterns are variablized expressions describing each target phenomenon, provided to the system prior to learning. 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.

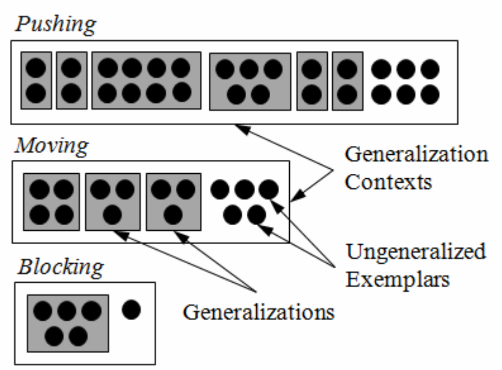

Figure 3: Generalization contexts after learning

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) such that if a 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 H of all correlated phenomena *p* are less than the binary entropy of *t*, or $H(P(p)) \leq H(t)$. The binary entropy function is the appropriate criterion to use because it measures information gain (i.e., low entropy implies high gain). Only decisive generalizations are parameterized into EHs.

Extracting causal models

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 EH.

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 or before 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 in hypothesizing causal relationships. For instance, in one generalization, movement starts *with* a pushing with P = 0.5, and starts *after* a pushing with P = 0.5. Consequently, movement is unlikely as a 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 5). 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 understand it by instantiating its encapsulated histories. For each EH, if its participants and 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 (Figure 4). The sketch shows a book on a table. 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.

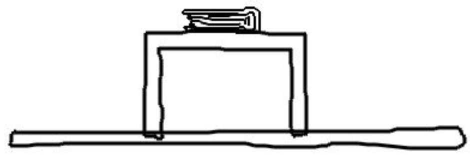

Figure 4: An example from Brown (1994)

Specifically, activating **`Block00`** to explain gravity pushing the book requires assuming two additional events, via the conditions in

Figure 5: (1) the gravity ?*P2* pushes the book ?*P3* in the direction *?dir1* of the initial push, and (2) the entity ?*P4* blocks the book ?*P3*. The table alone satisfies the constraints on *?P4*, binding the last of the non-event participants. This allows the simulation to assume 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 an EH, 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 tries to activate new EHs to explain it. The activation failures for those EH instances are provided as the reasons for the phenomenon not occurring, as shown below.

```
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

EXPERIMENT

To test whether this simulation can learn psychologically plausible models from multimodal stimuli, we examine 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 the human results, and then we describe the simulation setup and compare the 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 scientifically 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)
3. The table is not pushing or pulling (5)
4. The table is just blocking the book (4)
5. The book would move up if the table exerted a force (4)

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 horizontal 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.

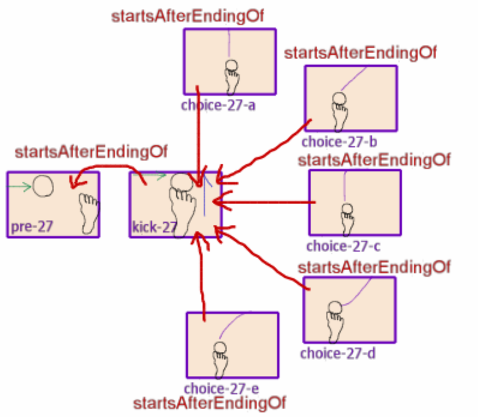

Figure 6: An example from the Force Concept Inventory (Hestenes et al, 1992)

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

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

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

Simulation setup

We implemented our simulation using the Companions Cognitive Systems architecture (Forbus et al., 2008). We used 16 sketches with accompanied natural language as learning stimuli, using examples motivated by the mental models literature cited earlier. Like Figure 2, all stimuli include pushing phenomena, and either movement or blocking phenomena. The learning stimuli did not include the test scenarios. The SEQL assimilation threshold was set to 0.6 and the EH probability threshold was set to 0.9. The temporal encoding step resulted in 25 pushing exemplars, 15 moving exemplars, and 6 blocking exemplars.

Because the SEQL model of analogical generalization is order-dependent, different orderings of learning stimuli may yield different generalizations, which in turn may produce different encapsulated histories. Consequently, the order in which learning stimuli are provided to the simulation could affect the simulation's behavior on the Brown (1994) and FCI reasoning tasks. We ran our simulation with 60 random orderings of the 16 multimodal stimuli. This is a very small sampling of the 21 trillion possible orderings, but it demonstrates that the order of stimuli can affect learning and reasoning.

As expected, varying the stimuli order affected the number and content of SEQL generalizations, the number and content of encapsulated histories, and the behavior on the reasoning tasks. The SEQL organization resulting after learning one of the 60 stimuli orderings is shown in Figure 3. Two of the resulting encapsulated histories from the same ordering are shown in Figures 1 and 5. We provide a more detailed analysis in the below.

Comparing Human and Simulation Results

How does the system behavior compare to human results? In each of the 60 trials we ran both reasoning tasks, and we discuss the results below.

The simulation's behavior on Brown's (1994) test scenario can be classified in one of five ways, as shown in Figure 7: (1) overspecific, (2) counterfactual explanation, (3) citing the book pushing the table, (4) providing both explanations 2 and 3, and (5) overgeneral. We discuss each classification in turn.

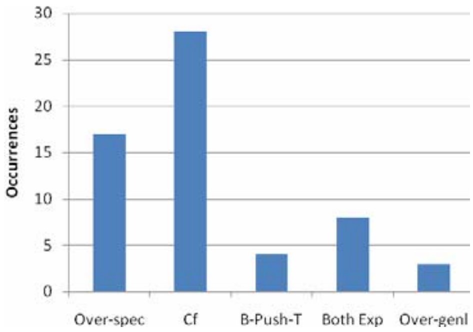

Figure 7: Simulation outcomes on Brown's task

In 28% of the trials, the simulation's concepts were overspecific – the system could not justify or refute the table's pushing against the book. This occurs when no encapsulated histories are activated during reasoning. While encapsulated histories were produced for all overspecific trials, they were too specific to be activated during reasoning: they contained extra participants or conditions that were not present in the test scenario.

In 47% of the trials, the simulation uses a counterfactual explanation by activating an EH like that of Figure 1 to make 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.

In 7% of the trials, the simulation answers that the book pushes against the table by finding activated EHs in which the book and table jointly participate to explain their behavior in the scenario. Consequently, it uses an EH similar to 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 4 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 that it could, but this is an empirical question.

In 13% of the trials, the system used both the counterfactual explanation as well as citing the book pushing the table. In these trials, the simulation learned both types of EHs (similar to figures 1 and 5) necessary to make both types of explanation. As noted in Brown (1994), some students provided multiple explanations to justify their answer.

In the remaining 5% of the trials, the system learned overly-general pushing models: pushing could occur without blocking or movement. Consequently, the system justified the table's push on the book and the book's push on the table with these over-general models. The statistical criterion for finding decisive generalizations failed in these trials: none of the generalizations within the *pushing* context were highly correlated with *blocking* or *movement*, so movement was not believed to be a necessary consequence of pushing. This allowed the simulation to reason about pushing without consequence, so while this behavior does not resemble a misconception from Brown (1994), it is also scientifically incorrect.

In the same 60 trials, the simulation's output on the FCI scenario can be classified in two ways. In 48% of the trials, 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, making it the most popular misconception. In the remaining 52% of the trials, the system is overly specific: all of the EHs learned are too specific to be activated in the FCI scenario.

The learned EHs were overly specific for the Brown problem about 25% of the time, and about 50% of the time for the FCI scenario. There were several statistical discrepancies between overspecific trials and trials that yielded explanations during problem solving. Overspecific trials had, on average, 10% more unassimilated exemplars, 20% less decisive generalizations, and twice as many indecisive generalizations. We address each of these factors next.

An unassimilated exemplar occurs when an exemplar is not similar to existing generalizations and other exemplars. They are not considered when building encapsulated histories, though they may contain useful data. With few exceptions, when exemplars are excluded from the generalizations, the generalizations are more specific. Having more *decisive* generalizations results in building more encapsulated histories for use during problem solving. Conversely, having more *indecisive* generalizations can result in gaps in conceptual knowledge: like unassimilated exemplars, these data are not used during problem solving. This is another factor in generating overspecific EHs.

Overall, the results of this experiment demonstrate that the models learned by the simulation are like those of pre-physics students. The simulation learned the same misconceptions as the majority of the students whose answers were not scientifically correct. Moreover, we have demonstrated that the order of stimulus presentation affects the number and content of models learned by the simulation.

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, but our system learns models of physical behaviors and causal laws, while SINS learns associations between environmental conditions and control parameters. Our particular method of multimodal stimuli encoding has been used by Lockwood et al. (2005) to model the learning of spatial prepositions.

DISCUSSION & FUTURE WORK

We have described how analogical generalization and qualitative modeling can be used to simulate the process of learning naive physics models and misconceptions. To reduce tailorability, the simulation inputs were combinations of sketches and simplified English. The answers given by the learned models match a subset of those given by human students on the same test problems, for many orders of stimulus presentation.

While we believe that this is a significant step, more work remains. We plan to expand the phenomena covered to include the entire FCI, for example. Can this simulation, with an expanded corpus, learn a correct model of forces and motion as well as cover the entire space of human misconceptions? Another important limitation of our model is that it is currently batch, whereas people build up models incrementally with experience. We plan on extending the simulation to operate incrementally, using a model of metacognition to detect and hopefully correct errors of over-generality and over-specificity. Finally, we plan to incorporate these ideas in a larger-scale learning model, where the quality and content of its predictions guide future learning. We have presented a model of how misconceived intuitive models can be learned from experience, which is the first step in 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. and 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. and 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*.

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