

Analogical model formulation for transfer learning in AP Physics

Matthew Klenk^{*,1}, Ken Forbus

Qualitative Reasoning Group, Northwestern University, 2133 Sheridan Road, Evanston, IL 60208, USA

ARTICLE INFO

Article history:

Received 29 September 2008

Received in revised form 4 September 2009

Accepted 18 September 2009

Available online 23 September 2009

Keywords:

Transfer learning

Analogical reasoning

Model formulation

Case-based reasoning

ABSTRACT

Transfer learning is the ability to apply previously learned knowledge to new problems or domains. In qualitative reasoning, *model formulation* is the process of moving from the unruly, broad set of concepts used in everyday life to a concise, formal vocabulary of abstractions, assumptions, causal relationships, and models that support problem-solving. Approaching transfer learning from a model formulation perspective, we found that analogy with examples can be used to learn how to solve AP Physics style problems. We call this process *analogical model formulation* and implement it in the Companion cognitive architecture. A Companion begins with some basic mathematical skills, a broad common sense ontology, and some qualitative mechanics, but no equations. The Companion uses *worked solutions*, explanations of example problems at the level of detail appearing in textbooks, to learn what equations are relevant, how to use them, and the assumptions necessary to solve physics problems. We present an experiment, conducted by the Educational Testing Service, demonstrating that analogical model formulation enables a Companion to learn to solve AP Physics style problems. Across six different variations of relationships between base and target problems, or *transfer levels*, a Companion exhibited a 63% improvement in initial performance. While already a significant result, we describe an in-depth analysis of this experiment to pinpoint the causes of failures. Interestingly, the sources of failures were primarily due to errors in the externally generated problem and worked solution representations as well as some domain-specific problem-solving strategies, not analogical model formulation. To verify this, we describe a second experiment which was performed after fixing these problems. In this second experiment, a Companion achieved a 95.8% improvement in initial performance due to transfer, which is nearly perfect. We know of no other problem-solving experiments which demonstrate performance of analogical learning over systematic variations of relationships between problems at this scale.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Transfer learning research is motivated by the observation that people improve in their ability to learn new domains based on their experiences in related tasks. We focus here on the task of *model formulation* [10]. Given a scenario description, a domain theory of model fragments, and a question, model formulation produces a *scenario model*, which consists of the relevant abstractions, processes, and causal relationships useful for answering the question. An important contribution of the qualitative reasoning community has been formalizing this process. For example, methods have been developed to efficiently identify what levels of detail should be included and which perspectives should be taken in a scenario model

* Corresponding author.

E-mail address: matthew.klenk.ctr@nrl.navy.mil (M. Klenk).

¹ Current address: Naval Research Laboratory, 4555 Overlook Ave SW, Washington, DC 20375, USA.

1. A ball is released from rest from the top of a 200m tall building on Earth and falls to the ground. If air resistance is negligible, which of the following is most nearly equal to the distance the ball falls during the first 4s after it is released? (a) 20m; (b) 40m; (c) 80m; (d) 160m.
2. An astronaut on a planet with no atmosphere throws a baseball bat upward from near ground level with an initial speed of 4.0m/s. If the baseball bat rises to a maximum height of 5.0m, what is the acceleration due to gravity on this planet? (a) 0.8m/s²; (b) 1.2m/s²; (c) 1.6m/s²; (d) 20m/s².
3. A box of mass 8kg is at rest on the floor when it is pulled vertically upward by a cord attached to the object. If the tension in the cord is 104N, which of the following describes the motion, if any, of the box? (a) It does not move; (b) It moves upward with constant velocity; (c) It moves upward with increasing velocity but constant acceleration; (d) It moves upward with increasing velocity and increasing acceleration.
4. A block of mass M is released from rest at the top of an inclined plane, which has length L and makes an angle q with the horizontal. Although there is friction between the block and the plane, the block slides with increasing speed. If the block has speed v when it reaches the bottom of the plane, what is the magnitude of the frictional force on the block as it slides? (a) $f = Mg\sin(q)$; (b) $f = Mg\cos(q)$; (c) $f = MgL\sin(q) - \frac{1}{2}Mv^2$; (d) $f = [MgL\sin(q) - \frac{1}{2}Mv^2]/L$.

Fig. 1. Example AP Physics problems of the four types used in this work.

[35,42]. However, these approaches have three limitations. First, they rely on having a complete and correct domain theory. Such domain theories are difficult to construct. A more incremental, learning-oriented approach would be valuable for many applications, so that a system's competence could be improved over time as needed. Second, work in model formulation tends to start with fairly abstract scenario descriptions, e.g. circuit schematics or process diagrams. While this is fine for engineering applications, the ability to create qualitative and quantitative models of everyday situations (e.g., the scenarios found in physics problems) is one of the hallmarks of human flexibility in problem-solving. Third, also due to an emphasis on engineering domains, model formulation research has largely ignored learning. We propose instead a quite different approach: *analogical model formulation*. Analogical model formulation builds scenario models of everyday situations, based on prior experience. We believe that analogical model formulation provides a way to create systems that can incrementally learn a domain by making effective use of what knowledge they have, even when it is incomplete.

Solving physics problems provides a good example of the need for this kind of flexibility. Fig. 1 provides four examples, illustrating types of problems that our system learns to solve. (These problems will be used as examples through the paper.) We factor out natural language understanding by using predicate-calculus versions of these problems, but, unlike previous systems such as MECHO [3] or ISAAC [36], the translation process leaves everyday concepts in place. That is, balls, buildings, astronauts, boxes, baseball bats, flying, falling, and pulling all appear in the formal problem descriptions.² Understanding the relevant abstractions and assumptions for a physics problem stated as an everyday scenario is a difficult problem. Modeling decisions are contextual. For example, a coin falling off a building can be considered to be a point mass. But if we were modeling the exact same coin spinning on a table, it cannot be considered a point mass since its shape and size must be considered. The generalizations in any common-sense ontology are unlikely to provide much help: cats, coins, and pianos can all be considered as point masses in particular situations, but they are not closely related in any non-trivial ontology we are aware of. Analogical model formulation addresses the three limitations in model formulation research outlined above. First, since it relies on examples, analogical model formulation does not require a complete domain theory. Second, it operates directly with representations of situations drawn from a broad vocabulary of concepts. Finally, by accumulating examples, a system using analogical model formulation learns to formulate new models of different situations.

While complex, there is ample evidence that people are able to solve physics problems stated in everyday terms. The problems used throughout this work were generated by the Educational Testing Service, which administers the AP Physics examination in the United States. The AP Physics exam tests the ability of high school students to solve physics problems. Students' performance on this exam indicates that they do learn to categorize everyday objects in terms of domain abstractions, determine what equations are relevant, infer parameter values from scenarios, and assume default circumstances when necessary. The problems used in this work were generated automatically, from templates. The four problems, one from each problem type, shown in Fig. 1 represent roughly 20% of the typical Mechanics portion of the AP Physics examination.

Solving physics problems via analogical model formulation begins by retrieving an example analogous to the current scenario. Analogical model formulation uses the explanation of this example to formulate a model of the current scenario. Finally, the system uses traditional rule based reasoning over the model to arrive at a solution for the problem. Using example explanations, analogical model formulation enables the system to learn from examples how to make the following modeling decisions necessary for solving physics problems:

- Which equations are relevant and how they should be instantiated (e.g., the force exerted on the box is equal to the mass of the box multiplied by the acceleration of the box).
- Which assumptions to make by default (e.g., assuming that events happen on Earth).

² We used a subset of the ResearchCyc ontology, containing over 30,000 concepts. See <http://research.cyc.com> for details.

- Which assumptions about the values of specific quantities to make based on the scenario (e.g., objects at rest have 0 m/s).

We implement analogical model formulation using the Companion cognitive architecture [18]. A central hypothesis of the Companion architecture is that the flexibility and breadth of human common sense reasoning arises from analogical reasoning and learning from experience [17]. That is, people use their experience to enable them to solve new problems, and over time, extract generalizations and heuristics. For model formulation, this is consistent with Falkenhainer's [8] observation that engineers often use analogy with their experience to create new models. Klenk et al. [28] showed that a Companion can formulate models by analogy to solve everyday physical reasoning problems, such as those on the Bennett Mechanical Comprehension Test [2]. This article goes beyond that result by demonstrating that analogical model formulation can be used to solve variations of AP Physics style problems, through an external evaluation involving a substantial number of problems over systematic variations in relationships between problems.

Characterizing how well learned knowledge transfers is complex. One way involves identifying different *transfer levels*, each representing a particular type of relationship between a known *source* problem and a novel *target* problem. We use an externally-developed set of six transfer levels³ in this research. To illustrate them, we use Problem 1 from Fig. 1 as an example of a source problem:

A ball is released from rest from the top of a 200 m tall building on Earth and falls to the ground. If air resistance is negligible, which of the following is most nearly equal to the distance the ball falls during the first 4 s after it is released?

1. *Parameterization*: Target problem has different parameter values, but the qualitative outcome is the same.

A ball is released from rest from the top of a 500 m tall building on Earth and falls to the ground. If air resistance is negligible, which of the following is most nearly equal to the distance the ball falls during the first 3 s after it is released?

2. *Extrapolation*: Target problem has parameter values that are so different that the qualitative outcome changes as well.

*A ball is released from rest from the top of an 80 m tall building on Earth and falls to the ground. If air resistance is negligible, which of the following is most nearly equal to the distance the ball falls during the first 5 s after it is released?*⁴

3. *Restructuring*: The target problem asks for a different parameter.

A ball is released from rest from the top of a 200 m tall building on Earth and falls to the ground. If air resistance is negligible, how long does it take to fall 80 m?

4. *Extending*: The target problem includes distracting information.

A ball with a mass of 5 kg is released from rest from the top of a 100 m tall building on Earth and falls to the ground. If air resistance is negligible, which of the following is most nearly equal to the distance the ball falls during the first 4 s after it is released?

5. *Restyling*: The target problem involves different types of everyday objects.

A crate is dropped off the edge of a 100 m cliff on Earth and falls to the ground. If air resistance is negligible, which of the following is most nearly equal to the distance the crate falls during the first 4 s after it is released?

6. *Composing*: The target problem requires combining concepts from two different base problems.

An astronaut on a planet with no atmosphere throws a ball upward from near ground level with an initial speed of 4.0 m/s. The ball rises to a maximum height of 5.0 m before returning to the astronaut who then drops the ball from the top of a 100 m tall building. If air resistance is negligible, which of the following is most nearly equal to the distance the ball falls during the first 4 s after it is released? (Composed the Source Problem with Problem 2 from Fig. 1.)

We describe how a Companion using analogical model formulation solves AP Physics style problems, across these six transfer levels. We start by briefly reviewing the key aspects of the Companion cognitive architecture and the representations used in this work. Next, we illustrate the model formulation challenges in AP Physics and highlight how analogy can solve

³ These levels are from a 10-level catalog of transfer tasks used in DARPA's Transfer Learning Program (http://www.darpa.mil/ipto/programs/tl/docs/TL_Brief.ppt).

⁴ In 5 s, a ball falling from rest would travel 125 m. However the building the ball falls from is only 80 m tall; therefore the correct answer is that the ball falls 80 m.

these problems. Then we describe the analogical problem-solving method, which learns by accumulating worked solutions. We discuss an experiment, administered by ETS, in which a Companion using analogical model formulation achieved a 63.8% initial improvement across the six transfer levels. We present a detailed analysis indicating that most problem-solving failures were caused by human errors in the implementation and representations, and not due to analogical model formulation. After addressing these issues, a second experiment was performed in which the Companion achieved an initial improvement of 95.8% averaged across the six transfer levels. We close with a discussion of additional related work and our plans to build upon these results.

2. Companion cognitive architecture

Through the Companion architecture, we are exploring the hypothesis that analogical processing [21,17] is central to human reasoning and learning. Forbus et al. [19] provide an overview of the theoretical commitments of the Companion architecture. We begin by outlining the agent architecture and how it supports analogical model formulation. Then, we describe the computational models for analogical matching and retrieval it uses, and how they facilitate transfer.

2.1. Agent architecture

The Companion architecture is a distributed agent architecture. Each agent has its own knowledge base (KB), whose contents are periodically updated and synchronized. Communication between agents occurs through KQML messages [11]. For these experiments, the following agents were used:

- *Session Manager*: Provides facilities for user interaction.
- *Facilitator*: Manages sessions and directs communications between agents.
- *Executive*: Monitors the Companion's responsibilities and delegates work to the appropriate agents (e.g. follows scripts describing experiments, records quiz results, checkpoints KBs, etc.).
- *Session Reasoner*: Performs domain reasoning, in this case physics problem-solving.
- *Similarity-based Retriever*: Monitors the working memory of the Session Reasoner, and provides similar prior cases when there is a close match.

The Session Manager runs locally on the user's machine, the rest of the agents run on cluster nodes. New problems are given either individually through the Session Manager, or by a script describing an experiment which is uploaded to the Executive. The Executive hands the problem to the Session Reasoner, which implements all but the retrieval portion of the analogical model formulation and problem-solving processes. While the MAC/FAC algorithm (see below) used in the Retriever is efficient, distributing it reduces the memory load on the Session Reasoner as the size of case libraries rises.

2.2. Computational models of analogical processes

We use Gentner's [21] structure-mapping theory, which postulates that analogy and similarity are based on a structural alignment between two representations (the *base* and the *target*). The alignment process constructs one or more maximal structurally consistent matches. A structurally consistent match is one that satisfies the constraints of *tiered-identity*, *parallel connectivity*, and *one-to-one mapping*. The tiered-identity constraint provides a strong preference for only allowing identical predicates to match, but allows for rare exceptions. For example, *minimal ascension* [7] allows non-identical predicates to match if they are part of a larger mapped structure and share a close common ancestor in the ontology, which is useful for cross-domain analogies. Analogical model formulation uses within-domain analogies between problems and examples. Therefore, in this work, only identical predicates are allowed to match. Parallel connectivity states that if two predicates are matched then their arguments must also match. The one-to-one mapping constraint requires that each element in the base corresponds to at most one element in the target and vice versa. To explain why some analogies are better than others, structure-mapping uses the principle of *systematicity*: Mappings that are highly interconnected and contain deep chains of higher order relations are preferred over mappings with an equal number of relations which are independent from each other. Such nested structures indicate explanations, which provide context to help evaluate analogical inferences.

The Structure-Mapping Engine (SME) [9] models analogical matching. Given two structured representations as input (the base and target), SME produces one or more *mappings*, each representing a construal of what items (*entities* and *expressions*) in the base go with what items in the target. Each mapping is represented by a set of *correspondences*. Mappings also include *candidate inferences* which are conjectures about the target using expressions from the base which, while unmapped in their entirety, have subcomponents that participate in the mapping's correspondences. Based upon the systematicity constraint, a *structural evaluation score* is computed to estimate the match quality. SME operates in polynomial time, using a greedy algorithm [15,20]. The heart of transfer is the extraction of knowledge from prior examples using SME's candidate inferences.

To illustrate this more concretely, let us examine SME's operation over the small base and target representations from Table 1. The base description contains six facts describing a ball being released and falling. The target description describes

Table 1
Example base and target descriptions.

Base “A ball is released and falls”	Target “A box is released”
Entities – Fall-1, Ball-1, Release-1	Entities – Box-1, Release-2
Expressions (a) (FallingEvent Fall-1) (b) (Ball Ball-1) (c) (ReleaseOfSupport Release-1) (d) (objectActedOn Release-1 Ball-1) (e) (causes-EventEvent Release-1 Fall-1) (f) (primaryObjectMoving Fall-1 Ball-1)	Expressions (g) (BoxTheContainer Box-1) (h) (ReleaseOfSupport Release-2) (i) (objectActedOn Release-2 Box-1)

a box getting released. An initial mapping is created by matching identical predicates in the expressions, (d) \leftrightarrow (i) and (c) \leftrightarrow (h). Because they are structurally consistent (i.e., the mapping does not violate the parallel connectivity or one-to-one mapping constraints), this mapping is accepted with the following correspondences: Release-1 \leftrightarrow Release-2 and Ball-1 \leftrightarrow Box-1. Partially matched expressions from the base become candidate inferences by replacing the parts of the base expression which participate in the correspondences with the target items. In this case, expression (b) becomes (Ball Box-1), expression (e) becomes (causes-EventEvent Release-2 (AnalogySkolemFn Fall-1)) and expression (f) becomes (primaryObjectMoving (AnalogySkolemFn Fall-1) Box-1). The expression AnalogySkolemFn is used to represent unmapped entities from the base. Candidate inferences are conjectures about the target. In this example, some are correct, i.e., the Release-2 event causes something like the Fall-1 event from the base, and the Box-1 is the object moving of this event. On the other hand, they need not be correct (e.g., Box-1 is not a Ball). This paper describes how analogical model formulation uses the correspondences and candidate inferences to solve physics problems based on examples.

MAC/FAC [16] models similarity-based retrieval. It takes as input a *probe* and a *case library*. The probe is a structured description, representing what is currently being worked on by some system. The case library is a set of cases, each a structured description, representing the set of available examples. MAC/FAC selects a case from the case library based upon similarity with the probe. It does this in two stages. The first stage (MAC) computes a special kind of feature vector for the probe and each case in the case library, whose components are proportional to the number of occurrences of individual predicates in each structured representation. The case from the case library with the highest (or up to three, if very close) dot product with the probe is returned from the MAC stage. The second (FAC) stage uses SME to compare these candidates to the probe. The candidate with the highest structural evaluation score is returned by the algorithm as its reminding. (If others are very close, up to three can be returned, but in this work, only the most similar reminding is used.)

Both SME and MAC/FAC have been used successfully in many domains (e.g. case-based coaching, reasoning about physical systems, and thermodynamics), and as cognitive simulations to model a number of psychological results [13]. Here, these domain independent processes facilitate transferring knowledge at each of the six transfer levels. Because SME and MAC/FAC focus on structural matches, numbers are treated as entities, whose specific values are ignored. Therefore the matching process is insensitive to particular numeric values (i.e., 5 is treated the same as 500), simplifying parameterization transfer. The emphasis on relational structure aids extrapolation and restructuring problems because contextual information in the base remains associated in the candidate inferences. Both SME and MAC/FAC handle partial matches, facilitating the handling of extending and restyling problems. Finally, composing, as explained below, is achieved by using multiple retrievals to cover all the phenomena mentioned in the problem.

3. Representations of problems and worked solutions

When students study for the AP Physics exam, one important way they learn is by doing problem sets. For feedback, students often get *worked solutions*. These step-by-step explanations are always used in textbooks to illustrate problem-solving. Worked solutions are typically incomplete, outlining steps abstractly rather than at the level of detail found in a proof. Our system is designed to use such worked solutions to formulate models of new problems. In collaboration with ETS and Cycorp, we developed representation conventions for problems and worked solutions. These conventions allowed us to factor out natural language understanding, while keeping the incomplete nature of worked solutions intact.

All of the representations used in this work are in CycL, the predicate calculus language of the ResearchCyc KB [33]. The KB used by the Companion’s agents consists of a subset of ResearchCyc KB, plus our own extensions. The extensions include problem-solving strategies and an implementation of QP theory [12], as well as rules for inferring some kinds of qualitative information from pre-existing ResearchCyc concepts. ResearchCyc is useful for representing physics problems and worked solutions because it includes over 30,000 distinct types of entities, over 8000 relationships and functions, and 1.2 million facts constraining them. Thus, the everyday concepts that appear in physics problems like “astronaut” and “dropping” are already defined for us, rather than us generating them specifically for this project. This reduces tailorability in our experiments. In addition to the templates used to create the problems in Fig. 1, ETS and Cycorp developed templates to generate problems and worked solutions representing each transfer level. Consequently, all the problems and worked

```

...
(isa Throwing-1 ThrowingAnObject)
(isa Astronaut-1 Astronaut)
(isa Planet-1 Planet)
(isa BaseballBat-1 BaseballBat)
(groundOf Planet-1 Ground-1)
(performedBy Throwing-1 Astronaut-1)
(no-GenQuantRelnFrom in-ImmersedFully Planet-1 Atmosphere)
(eventOccursNear Throwing-1 Ground-1)
(objectThrown Throwing-1 BaseballBat-1)
(querySentenceOfQuery Query-1 (valueOf (AccGravityFn Planet-1) Acc-1))

```

Fig. 2. Part of the representation of Problem 2 (simplified for readability).

solutions in this evaluation were created externally. The representations of the 460 physics problems⁵ used in this evaluation contained 4973 instances from 108 conceptual types and 103 unique relations. When including the worked solutions, the representations include 11,230 instances from 110 types and 144 relations.

3.1. Example problem and worked solution

The problem representations are intended to be direct translations into predicate calculus from natural language problem statements, without any abstraction or reasoning. Fig. 2 shows a subset of the 37 facts used to represent Problem 2 from Fig. 1. The facts in Fig. 2 define the planet with no atmosphere, the astronaut throwing the bat and the question asking for the gravitational force of the planet. The average number of facts for each problem is 44.

The worked solutions are predicate calculus representations of the example problems found in textbooks. They are not deductive proofs, nor problem-solving traces of the operations of our solver. They leave out many steps and characterize problem-solving operations in very general ways. Here is an English rendering of the worked solution for Problem 2 from Fig. 1:

1. Categorize the problem as a distance–velocity problem under constant acceleration.
2. Instantiate the distance–velocity equation specific to the quantities of this problem (e.g. the bat and the upward motion event) ($V_f^2 = V_i^2 + 2ad$).
3. Given the projectile motion of the bat, lack of atmosphere, and the maximum altitude of bat, infer that the acceleration of the bat is equal to the acceleration due to the gravity of the planet ($a = g$), the distance the bat travels during the upward motion event ($d = 5$ m) and that the bat is not moving at the maximum height ($V_f = 0$ m/s).
4. Use the assumed values and the given parameters to solve the equation for the acceleration due to gravity ($g = -1.6$ m/s²).
5. Determine if there is a relevant boundary condition, i.e., ascertain that the answer is consistent ($g = -1.6$ m/s²).
6. Select the appropriate multiple choice answer (c).

The predicate calculus version of this worked solution consists of 104 facts.

Fig. 3 shows part of the representation for Step 3. The first fact indicates that this is an assuming value step. The `stepUses` statements give the context for assuming the values. The subset of `stepUses` statements displayed here state that there is no atmosphere on the planet, the throwing event occurs near the ground and that the bat is the object moving in the upward movement event. The last three facts contain the results of this step, which are values for specific parameters: the speed of the bat at the end of the upward movement event, the distance that bat travels during this event and the bat's acceleration during this event. Figs. 2 and 3 as well as the rest of the figures which include predicate calculus representations use simplified entity, predicate and function names to improve readability. The complete representations for the problem and worked solution of Problem 2 appear in Appendix A. The average number of facts across all the worked solutions is 163.

4. Analogical model formulation

The primary contribution of this work is the process of analogical model formulation. Our analysis of physics problems indicates that successful problem-solving typically requires four types of modeling decisions: relevance reasoning, quantity value assumptions, default circumstances, and modeling abstractions. This section describes each in turn, and how we use analogous worked solutions to make modeling decisions in new problems, without needing a complete domain theory.

Relevance reasoning in physics problem-solving determines which equations are applicable for a given situation. Even in a relatively constrained domain like AP Physics, the number of potentially relevant equations can be quite large, due to

⁵ The representations of these problems and worked solutions can be found at http://www.qrg.northwestern.edu/analogy_challenge/ap_physics.html.

```

(stepType Step3 DeterminingValueFromContext)
(stepUses Step3 (isa Throwing-1 ThrowingAnObject))
(stepUses Step3 (occursNear Throwing-1 Ground-1))
(stepUses Step3
  (no-GenQuantRelnFrom in-ImmersedFully Planet-1 Atmosphere))
(stepUses Step3 (objectMoving Upward-1 BaseballBat-1))
(stepUses Step3 (direction Upward-1 Up-Directly))
...
(stepResult Step3
  (valueOf
    (AtFn ((QPQuantityFn Speed) BaseballBat-1) (EndFn Upward-1))
    (MetersPerSecond 0)))
(stepResult Step3
  (valueOf ((QPQuantityFn DistanceTravelled) BaseballBat-1 Upward-1)
    (Meter 3)))
(stepResult Step3
  (valueOf (AtFn ((QPQuantityFn Acceleration) BaseballBat-1) Upward-1)
    ((QPQuantityFn AccelerationDueToGravity) Planet-1)))

```

Fig. 3. Problem 2 worked solution Step 3 (simplified for readability).

specialized forms. For example, while solving Problem 2, it would be a mistake for a system to consider magnetic forces on the baseball bat. Efficient problem-solvers must first determine which equations are relevant. Our method uses the insight that similar problems are likely to involve similar equations. All the equations for physics phenomena applied to a problem are found by searching the candidate inferences produced by the analogy between the new problem and worked solution(s).

Quantity value assumptions occur when the problem-solver infers a parameter value from the scenario. For instance, in Problem 2, the problem-solver must recognize that the distance the baseball bat travels is 5 m and that its velocity at the end of the upward motion event is 0 m/s. Neither of these facts is given explicitly in the problem. While the velocity at the end of the upward motion event could be derived via calculus or a qualitative model, the distance the bat travels is necessarily an approximation, because the scenario description states that the throwing event occurs “near ground level” and the maximum altitude of the bat is 5 m. Our method uses candidate inferences to suggest quantity values via analogy.

Physics problems frequently require problem-solvers to assume certain circumstances by default. The most common of these in AP Physics is to assume that events happen on Earth and are subject to Earth’s gravity. For example, Problem 3, the lifting box problem, requires this assumption to determine the net force on the box. Again, our method relies on analogy to find such default circumstances.

The last type of modeling decision involves categorizing everyday objects as abstractions. When reasoning with a domain theory defined in abstract terms, it is necessary to move from the everyday objects and events to this abstract vocabulary. This is another form of relevance reasoning, because abstractions are a way of framing the problem in terms of what phenomena should be considered. Given the problem of a ball falling off the building, a problem-solver would likely abstract the ball into a point mass and not an electrical particle, thus pruning the search space to the appropriate equations and relevant assumptions. As indicated above, the relevant equations and assumptions are suggested via analogy. In other words, abstraction modeling decisions are implicit in the other modeling decisions made by analogical model formulation.

Here we show how a Companion uses analogical model formulation to make these modeling decisions for the following restyling variation of Problem 2 from Fig. 1:

“A physicist on an asteroid with no atmosphere throws a spoon upward from near ground level with an initial speed of 4.0 m/s. If the spoon rises to a maximum height of 5.0 m, what is the acceleration due to gravity on this asteroid? (a) 0.8 m/s²; (b) 1.2 m/s²; (c) 1.6 m/s²; (d) 20 m/s².”

First, the Retriever, using MAC/FAC, provides the worked solution to Problem 2 (outlined in Section 3.1) as a reminding. Then, the Session Reasoner uses SME to create an analogy with this reminding as the base and the new problem as the target. The most relevant correspondences from the best mapping are summarized in Table 2. Recall that candidate inferences are expressions from the base (here, the worked solution) that are conjectured to hold in the target, by virtue of the mapping’s correspondences. A number of these are `stepUses` or `stepResult` statements, representing worked solution steps and their contexts which suggest modeling decisions in the problem. Analogical model formulation draws upon these candidate inferences to incrementally build a scenario model for the problem.

As an example of relevance reasoning, Step 2 of the worked solution contains the equation $V_f^2 = V_i^2 + 2ad$ in terms of the baseball bat and its upward movement event in the original problem. The candidate inferences generated from this step include a corresponding equation with the quantities V_f , V_i , a , and d in terms of the problem entities: `Spoon-5` and `Upward-5`.

Analogical model formulation handles decisions concerning quantity value assumptions and default circumstances in the same manner. Worked solutions contain steps indicating one of these assumptions was necessary. These steps appear as

Table 2

Correspondences between the worked solution and the problem scenario. Only correspondences used in creating candidate inferences are included, for brevity.

Worked solution item	Problem scenario item
Planet-1	Asteroid-5
BaseballBat-1	Spoon-5
Astronaut-1	Physicist-5
Upward-1	Upward-5
(StartFn Upward-1)	(StartFn Upward-5)
(EndFn Upward-1)	(EndFn Upward-5)

Given: a problem, P , and a case library, C , of worked solutions, $\{ws_1 \dots ws_n\}$

1. Retrieve analogue ws_i using MAC/FAC with probe: P and case library: C
 - 1.1. While there is unmapped event structure,
 - 1.1.1. retrieve additional analogue ws_j using MAC/FAC with probe: $(P - \{\text{facts covered by the currently mapped structure}\})$ and case library: $(C - \{\text{already retrieved worked solutions}\})$
2. Analyze P to determine the sought after quantity, q , and problem type, t .
3. Using analogical model formulation, solve for q by one of the following methods, using t to determine when the answer is appropriate:
 - 3.1. If the value of q is given in P , use that value.
 - 3.2. If the value of q is mentioned in a candidate inference, use that candidate inference.
 - 3.3. Search candidate inferences generated by analogs for an equation, e , mentioning q
 - 3.3.1. Recursively solve for each of the other quantities, $q_1 \dots q_i$, mentioned in e
 - 3.3.2. Solve e for q
4. If boundary condition check exists in candidate inferences,
 - 4.1. Compare mapped values in P to determine correct value for q
5. Use the computed value for q to select the multiple choice answer for P

Fig. 4. Solving AP Physics problems via analogical model formulation.

candidate inferences in the problem as a result of the analogical mapping. These candidate inferences suggest quantity value and default circumstance assumptions in the problem. In this mapping, the candidate inferences suggest that the velocity of the spoon at the end of its upward movement is zero, the distance the spoon travels during the upward movement event is 3 meters, and the acceleration of spoon during the upward movement event is the acceleration due to gravity of the asteroid. While default circumstances do not occur in this example, they are represented in the same way in worked solutions and handled by analogical model formulation in the same manner.

We wanted to provide the most stringent test of analogical model formulation from examples that we could. Consequently, the only modeling knowledge the system has concerns heuristics for evaluating candidate inferences. Importantly, the system has no general knowledge of physics equations, quantity values, or default circumstances. Without a reminding, it cannot solve any problems. This resulted in a useful simplification: the system does not explicitly categorize everyday objects in terms of abstractions. Making such abstractions explicit is useful only when there is abstract domain knowledge that will trigger on it. Such information is implicit in the choice of equations, quantity value assumptions, and default circumstances. As the experiments below indicate, this works well when the analogous problems are sufficiently similar. We expect explicit categorization to become important in more distant transfer, and indeed have conducted preliminary experiments in learning such knowledge [27]. However, since it is not necessary in these problems, we defer further discussion of this until Section 8.

5. Solving AP Physics problems via analogical model formulation

Here we describe our algorithm for solving AP Physics style problems, including analogical model formulation. Fig. 4 outlines the algorithm. After describing each step in detail, we outline how it is implemented in the Companion cognitive architecture.

5.1. Step 1: Retrieve analogous worked solutions

The process of solving a physics problem begins by generating an analogy with one or more relevant worked solutions. With the problem as the probe, MAC/FAC is used to retrieve a relevant example from a case library of worked solutions. The mapping between the worked solution (as the base) and the problem (as the target) is evaluated for adequacy by the loop in Step 1.1. Fundamentally, physics problems are about events. The *event structure* of a problem consists of the events

that occur in it. If the analogy does not map all of the event structure, additional analogues must be retrieved. Otherwise, there would be no knowledge from which to formulate a model for the unmapped events.⁶ For each iteration in Step 1.1, the already-matched parts of the probe are suppressed, so that retrievals are focused only on cases that are similar to the unmatched aspects of the problem. This was essential for handling the Composing transfer condition, since multiple analogues are needed to solve a single problem. For example, solving the Composing example in Section 1 might retrieve a worked solution for Problem 1 of Fig. 1, which would cover the release and falling events, but not the throwing and upward motion events. This would cause another retrieval to be made using MAC/FAC, with only the facts pertaining to the throwing and upward motion events as the probe. These two retrievals result in two sets of candidate inferences, both of which are available to the problem-solver in the following steps.

5.2. Step 2: Problem analysis

Solving most physics problems eventually boils down to finding the value for some quantity. But which quantity, and what form of description is appropriate for the value, must be ascertained by analyzing the problem. There are several different broad types of problems on the AP Physics exam. The subset of the exam used in this work contains the following types of problems:

- **Numeric value problems:** Determining the numeric value of a specific parameter.

A ball is released from rest from the top of a 200 m tall building on Earth and falls to the ground. If air resistance is negligible, which of the following is most nearly equal to the distance the ball falls during the first 4 s after it is released? (a) 20 m; (b) 40 m; (c) 80 m; (d) 160 m.

- **Symbolic value problems:** Determining the symbolic value of a specific parameter.

A block of mass M is released from rest at the top of an inclined plane, which has length L and makes an angle q with the horizontal. Although there is friction between the block and the plane, the block slides with increasing speed. If the block has speed v when it reaches the bottom of the plane, what is the magnitude of the frictional force on the block as it slides? (a) $f = Mg \sin(q)$; (b) $f = Mg \cos(q)$; (c) $f = MgL \sin(q) - 1/2Mv^2$; (d) $f = [MgL \sin(q) - 1/2Mv^2]/L$.

- **State elaboration problems:** Determining which parameter value will produce a described outcome.

Which of the following tensions is required to move a box of mass 8 kg from rest on the floor upward with constant acceleration when it is pulled vertically upward by a cord attached to the box? (a) 40 N; (b) 60 N; (c) 70 N; (d) 120 N.

- **Qualitative behavior problems:** Determining the qualitative outcome of a situation.

A box of mass 8 kg is at rest on the floor when it is pulled vertically upward by a cord attached to the box. If the tension in the cord is 104 N, which of the following describes the motion, if any, of the box? (a) It does not move; (b) It moves upward with constant velocity; (c) It moves upward with increasing velocity but constant acceleration; (d) It moves upward with increasing velocity and increasing acceleration.

This step identifies the problem type and sought quantity by analyzing the facts describing the query of the problem and the multiple choice answers. If the query concerns a quantity, then that is considered to be the sought quantity. In that case, the problem type is determined to be numeric or symbolic based on the kinds of expressions found in the possible answers. Instead of asking for specific quantity values, the query can concern a qualitative state. In these cases, if the possible answers are quantity values then the problem is a state elaboration problem, otherwise the problem is a qualitative behavior problem. For state elaboration problems, the sought quantity is determined by analyzing the event structure in the problem. In the example above, the acceleration of the box is the sought parameter. For qualitative behavior problems, the sought quantity is found by domain-specific rules that determine what value(s) are needed to distinguish between the possible answers. In the example above, for instance, the acceleration and velocity of the box during the pulling event would be sought.

The basic problem-solving strategy in physics problems is solving equations to find values. For numeric and symbolic value problems, this is sufficient. For qualitative behavior problems, the values of the sought quantities are tested to see which of the qualitative descriptions they satisfy. For state elaboration problems, we create an *assumption case* for each answer choice. The assumption case includes all the facts of the problem and an assumption of the value for the parameter

⁶ The distracters added in the Extending transfer condition never included events, only quantities and entities. Distracting events would cause the system to retrieve additional worked solutions to potentially model them. The effect of this on problem-solving performance would likely be limited to extending the length of time it took the Companion to produce the solution.

given in the answer choice (e.g., the tension in the example problem above). Then, for each assumption case, we solve for the sought quantity and determine if it is consistent with the problem description. If it is, then that assumption case is the correct answer.

5.3. Step 3: Solve for q via analogical model formulation

This step creates a scenario model incrementally, based on the analogy with the worked solution(s). The process starts by trying to find an appropriate value for the sought quantity q . In general, this is a recursive process, so we describe the general strategy for solving quantities.

Given a quantity q to be solved for, its value can be determined in one of three ways:

1. It is already known as part of the problem. That is, there is a `valueOf` statement in the problem representation that provides an appropriate value for q . By appropriate, we mean that when a numeric answer is sought, the `valueOf` statement provides a numeric value, and when a symbolic answer is sought, the `valueOf` statement is expressed in symbolic terms compatible with the possible answers, as ascertained in the previous section. In this case, the value from the statement is used.
2. It is assumable. That is, there is a candidate inference containing a `stepResult` statement which provides a value for q . In this case the value from the analogy is assumed.
3. It is mentioned in a relevant equation. That is, there is a candidate inference which contains an equation that mentions q . In this case, recursive subgoals are spawned to solve for the other quantities in the equation, and once their values are found, a value is derived for q .

While the first case is straightforward, the second and third cases make important modeling decisions via analogy. The second case handles quantity value assumptions and default circumstances. The third case handles relevance reasoning, since analogous situations are assumed to be governed by similar equations.

Analogical modeling decisions, like all non-deductive inferences, should be verified if possible [6]. The `stepUses` statements in the worked solution provide context for the worked solution step. These statements can be thought of as preconditions for the analogical modeling decision. Currently, we only use these preconditions in one situation. If these statements mention a planetary body, which is not included in the mapping, and there is a different planetary body in the problem, the analogical modeling decisions based upon this solution step are deemed unusable. This is a useful heuristic for this domain because decisions based on planetary bodies typically involve assumptions involving gravitational constants, which of course vary across planets. Currently, the rules that do these verifications are hand coded. In future work, we plan to enable Companions to learn and refine these rules with experience.

In addition to verifying the inference, it is important to understand how numbers are handled in the analogical mapping. Because these are all within-domain analogies, when there is a correspondence between number entities, it is likely spurious. For example, if the problem includes a ball moving at 1 m/s and the worked solution includes a ball moving at 2 m/s, then 1 could be placed in correspondence with 2. This is a spurious correspondence because there is no reason to believe that all 2's in the worked solution should be considered 1's in the problem. Therefore when candidate inferences for equations include numeric values, we use the number from the worked solution. When the candidate inferences concern an assumed value, the target value is used only when the units match; otherwise, the base value is used. Returning to our example, if the worked solution includes the distance-velocity equation, $V_f^2 = V_i^2 - 2ad$, then the resulting mapping would include a candidate inference suggesting $V_f^2 = V_i^2 - 1ad$ as an appropriate equation for the problem. Because we always use the number from the worked solution, even with the spurious correspondence, the correct equation $V_f^2 = V_i^2 - 2ad$ is instantiated in the problem.

Because the focus of this work is on model formulation, we provided our system with complete knowledge of units and conversions. The equation solving and algebraic simplification routines are straightforward, based on [14].

5.4. Step 4: Checking boundary conditions

Doing “sanity checks” of answers is always a good problem-solving practice. In physics, this involves testing boundary conditions. For example, if a problem asked, “How far a ball would fall off a 200 m building in 4 s?”, its worked solution would include a sanity checking step in which the computed answer, 80 m, was compared to the height of the building, 200 m. Since this is less, the computed answer is okay. If the computed answer were larger than the height of the building, it means that the boundary conditions of the equations are violated. Since one ignores the impact crater in these problems, the answer would then be the height of the building, because that is the point at which the behavior captured by the falling event ends.

This aspect of the scenario model also depends on the analogy. Boundary conditions are recognized by candidate inferences involving ordinal relationships (i.e., `greaterThan` or `lessThan`) between parameters in the problem. Currently

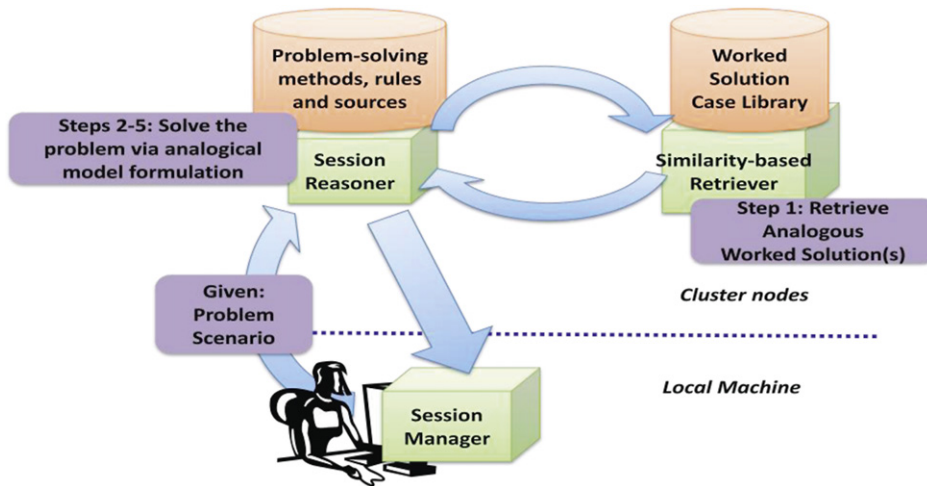


Fig. 5. Analogical model formulation implemented on the Companion cognitive architecture.

only boundary condition tests involving the sought quantity are processed. This is because it is clear how to resolve such a failure, i.e., use the value compared against it instead, because it constitutes a limit point [12] for that quantity.⁷

5.5. Step 5: Selecting the multiple choice answer

Finally, the appropriate multiple choice answer is selected. For numeric and symbolic value problems, the computed answer is compared against each of the answer choices and the closest answer is selected.⁸ For qualitative behavior problems, qualitative arithmetic is used to select the consistent answer choice. For instance, if there is a computed positive vertical velocity, the object must be moving upwards. In the example qualitative behavior problem of Section 5.2, the Companion determines that answer (c), the box moves upward with constant acceleration and increasing velocity, is the only consistent choice. This is because the box's velocity at the beginning of the event is 0 m/s and its computed acceleration during the event is 3 m/s². For state elaboration problems, the first assumed value that is consistent with the computed answer is selected. In the state elaboration example from Section 5.2, the problem states that the box is moving upward with constant acceleration, therefore, the consistent assumption case results in a positive acceleration for the box. The answer (d) contains the only tension, 120 N, which results in a positive acceleration, 5 m/s².

Importantly, the system is not allowed to guess if it cannot compute the answer for a problem.

5.6. Implementation in a Companion

Fig. 5 shows how the steps of the algorithm are divided among the agents in a Companion. Aside from retrieving worked solutions, the entire process takes place on the Session Reasoner. The Session Reasoner requests relevant worked solution(s) from the Similarity-based Retriever. When the Session Reasoner selects an answer, it is sent to the Session Manager for display on the user's machine. We implement the algorithm from Fig. 4 using an AND/OR problem-solver drawn from [14]. The problem-solving knowledge consists of 27 methods, 169 backchaining rules, and two reasoning sources, which are procedural attachments efficiently implementing analogical processing and algebraic operations.

For illustration, here is how a Companion employs the above algorithm to solve the following restyling problem:

"A box is dropped from the top of a 300 m cliff on Earth and falls to the ground. If air resistance is negligible, which of the following is most nearly equal to the distance the box falls during the first 7.3 s after it is released? (a) 36.5 m; (b) 73 m; (c) 266.45 m; (d) 532.9 m."

The problem is presented to the Companion as a case of 28 predicate calculus facts. The Companion begins by asking the Retriever for a relevant example, which in this case is the worked solution for Problem 1 from Fig. 1. The Session Reasoner uses SME create a mapping between the retrieved worked solution and the problem. The event structure of the problem contains three events: the initial situation, the dropping, and the falling. All three events are included in the correspondences of this mapping; therefore the Companion does not retrieve additional analogues. Next, it determines that this is a numeric

⁷ This heuristic is reasonable for mechanics but would not be appropriate for other domains, such as thermodynamics.

⁸ Learning to recognize when a computer answer is not "close enough" to any of the choices requires generalizing across experiences from the domain. Analogical model formulation focuses on the direct application of examples and consequently learning such strategies is outside the scope of this work.

<p>Base expressions:</p> <ol style="list-style-type: none"> 1. (valueOf ((QPQuantityFn DistanceTravelled) Ball-1 Interval-1) Distance-1) 2. (valueOf ((QPQuantityFn Time-Quantity) Interval-1) (SecondsDuration 4)) 3. (temporallyCooriginating Fall-1 Interval-1) 4. (TimeInterval Interval-1) 5. (objectActedOn Release-1 Ball-1) 6. (primaryObjectMoving Fall-1 Ball-1) 7. (objectStationary Initial-Situation-1 Ball-1) <p>Target Expressions:</p> <ol style="list-style-type: none"> 1. (valueOf ((QPQuantityFn DistanceTravelled) Box-5 Interval-5) Distance-5) 2. (valueOf ((QPQuantityFn Time-Quantity) Interval-5) (SecondsDuration 7.3)) 3. (temporallyCooriginating Fall-5 Interval-5) 4. (TimeInterval Interval-5) 5. (objectActedOn Release-5 Box-5) 6. (primaryObjectMoving Fall-5 Box-5) 7. (objectStationary Initial-Situation-5 Box-5)

Fig. 6. Expressions in alignment based upon identical predicates.

<p>Base Expression</p> <pre>(stepResults Step-5 (equationForSolution WorkedSolution-1 (mathEquals ((QPQuantityFn DistanceTravelled) Ball-1 Interval-1) (PlusFn (TimesFn (AtFn ((QPQuantityFn Speed) Ball-1) (StartFn Interval-1)) ((QPQuantityFn Time-Quantity) Interval-1)) (TimesFn 0.5 (AtFn ((QPQuantityFn Acceleration) Ball-1) Interval-1) (SquaredFn ((QPQuantityFn Time-Quantity) Interval-1))))))</pre> <p>Candidate Inference</p> <pre>(stepResults (AnalogySkolemFn Step-5) (equationForSolution (AnalogySkolemFn WorkedSolution-1) (mathEquals ((QPQuantityFn DistanceTravelled) Box-5 Interval-5) (PlusFn (TimesFn (AtFn ((QPQuantityFn Speed) Box-5) (StartFn Interval-5)) ((QPQuantityFn Time-Quantity) Interval-5)) (TimesFn (AnalogySkolemFn 0.5) (AtFn ((QPQuantityFn Acceleration) Box-5) Interval-5) (SquaredFn ((QPQuantityFn Time-Quantity) Interval-5))))))</pre>
--

Fig. 7. Candidate inference projected from the base due to correspondences between Ball-1 ↔ Box-5; Interval-1 ↔ Interval-5.

value problem based upon the query statement and the answer choices. The query statement for this problem indicates that the distance the box travels over the 7.3 second interval is the sought quantity.

Next, the Companion proceeds to solve for the sought quantity, the distance the box travels. Because there are neither `valueOf` statements concerning the distance the box travels nor candidate inferences suggesting a quantity value or default circumstance modeling assumption, the Companion searches for a relevant equation mentioning the sought quantity. This is done by searching the candidate inferences of the analogy. Fig. 6 contains the aligned expressions which result in the entities `Ball-1` and `Interval-1` from the worked solution corresponding with `Box-5` and `Interval-5` from the problem. These correspondences result in a candidate inference for the applicable equation mentioning the distance `Box-5` travels during `Interval-5`, $d = v_i t + .5at^2$, shown in Fig. 7. The candidate inference copies over the relational structure, substituting entities based on the correspondences. The `AnalogySkolemFn` expressions represent entities which appear in the base (i.e., worked solution) representation and do not have a corresponding entity in the target (i.e., problem description) representation. To use the equation suggested by this candidate inference, the Companion extracts the `mathEquals` subexpression and removes the `AnalogySkolemFn` from the `.5`.

In order to solve this equation for the distance the box travels, the Companion first solves for each of the other parameters in the equation: the duration of the interval, the speed of the box at the start of the interval, and the acceleration of the box during the interval. First, the duration of the `Interval-5`, 7.3 s, is given directly in the problem. Second, using the analogy to make a quantity value assumption, the Companion infers the speed of the box at the start of the interval, 0 m/s, based upon Step 5 of the worked solution. Step 5 states that `Ball-1` at the beginning of `Interval-1` has a speed of 0 m/s because it starts at rest. A subset of the candidate inferences used in this step are shown in Fig. 8. Because our algorithm uses the value from the base description, we simply remove the `AnalogySkolemFn` from 0 m/s when making

```
(stepUses (AnalogySkolemFn ETS-WorkedSolution-1-0-1-Step5)
  (primaryObjectMoving Fall-5 Box-5))
(stepUses (AnalogySkolemFn ETS-WorkedSolution-1-0-1-Step5)
  (objectStationary Initial-Situation-5 Box-5))
(stepUses (AnalogySkolemFn ETS-WorkedSolution-1-0-1-Step5)
  (contiguousAfter Fall-5 Initial-Situation-5))
(stepUses (AnalogySkolemFn ETS-WorkedSolution-1-0-1-Step5)
  (temporallyCooriginating Fall-5 Interval-5))
(stepResult (AnalogySkolemFn ETS-WorkedSolution-1-0-1-Step5)
  (valueOf
    (MeasurementAtFn ((QPQuantityFn Speed) Box-5)
      (StartFn Interval-5)
      (AnalogySkolemFn (MetersPerSecond 0))))))
```

Fig. 8. Candidate inferences permitting the quantity value modeling assumption for the speed of the box.

```
(stepUses (AnalogySkolemFn ETS-WorkedSolution-1-Step7)
  (greaterThanOrEqualTo (Meter 300) (AnalogySkolemFn (Meter 80))))
(stepUses (AnalogySkolemFn ETS-WorkedSolution-1-Step7)
  (valueOf (MeasurementAtFn ((QPQuantityFn Height) Cliff-5) Fall-5)
    (Meter 300)))
(stepUses (AnalogySkolemFn ETS-WorkedSolution-1-Step7)
  (valueOf ((QPQuantityFn DistanceTravelled) Box-5 Interval-5)
    (AnalogySkolemFn (Meter 80))))
```

Fig. 9. Candidate inferences indicating a boundary check condition.

this inference. This is reasonable because we are performing within-domain analogies; for cross-domain analogies, more work would be required to resolve the analogy skolem into an appropriate constant for the target domain. Next, this inference must be verified. The Companion makes sure that the step does not rely upon the worked solution occurring on a different planetary body. The `stepUses` statements, the context of the worked solution step, do not include references to any planetary bodies as defined by the `ResearchCyc` ontology. Therefore, the Companion assumes 0 m/s as the speed of the box at the start of the interval.

Third, the Companion makes a default circumstance modeling decision to determine that the acceleration of the box during the falling event is 10 m/s^2 . This decision is made in the same manner as the quantity value decision for the speed of the box at the beginning of the falling event, with one exception. The `stepUses` statements for the worked solution step suggesting this decision mention `PlanetEarth`, a planetary body. Because `PlanetEarth` is unmapped in the analogy, the Companion searches for any planetary bodies in the problem. The Companion accepts the inference because the problem does not mention any planetary bodies. This is an example of the Companion making a default circumstance modeling assumption. After recursively solving for these three parameters, the Companion solves the equation for the distance `Box-5` traveled during `Fall-5`, 266.45 m.

Next, the Companion checks for a candidate inference concerning a boundary condition check. In this case, Step 7 from the worked solution is a boundary condition check comparing the computed distance `Ball-1` fell, 80 m, against the height of `Building-1`, 200 m. Recall that candidate inferences substitute subexpressions based on the correspondences of the mapping. Fig. 9 contains the resulting candidate inferences indicating that the height of `Cliff-5` should be compared against the distance `Box-5` traveled. The computed answer, 266.45 m, is less than the height of `Cliff-5`, 300 m, and therefore the Companion uses the computed answer when selecting the multiple choice option.

In the final phase of problem-solving, the Companion uses the computed answer to select the appropriate multiple choice option. Because this is a numeric problem, the answer choice closest to the computed answer, 266.45 m, is selected. In this case, answer (c) is 266.45 m exactly, and, therefore, the Companion selects it. Solving this problem takes the Companion approximately 35 seconds.⁹

6. Evaluation

Two experiments were conducted to evaluate a Companion's ability to transfer knowledge across physics problems via analogical model formulation. The first experiment was an external evaluation conducted by the Educational Testing Service on largely unseen problems. The timing of this evaluation was determined by an external, funder-mandated timetable and code freeze. The results of this evaluation were summarized in [24], but we provide additional details and analysis here.

⁹ The Companion was running on 4 cluster nodes, each with two 3.2 GHz Pentium Xeon processors and 3 GB of RAM.

Although the results, as described below, showed significant transfer across all six levels, we were surprised that they were not better. An analysis of these results led to the discovery of numerous representation and implementation errors. After fixing these bugs, a second experiment, with the same design as the first, was run. This experiment confirms that the problems were not with analogical model formulation, and provides evidence about the asymptotic performance of analogical model formulation. This section discusses both experiments in turn, including our post-analysis and changes made to the system and representations before the second experiment.

6.1. Experiment 1

The first evaluation was carried out by the Educational Testing Service, who remotely accessed a Companion running on our cluster. The evaluation materials and methodology were designed by ETS, without detailed knowledge of our learning method, analogical model formulation. The problems and worked solutions in this experiment were generated from the templates, examples from four of which appear in Fig. 1. These templates represent approximately 20% of the typical Mechanics portion of the AP Physics exam. The authors saw examples from less than 50% of the templates before the evaluation, and none of the templates themselves. ETS measured the amount of transfer learning in our system. Transfer learning occurs when an agent learns faster in the target domain given learning in a related source domain. This is most naturally measured as a difference in learning curves, which motivates this experiment's design.

6.1.1. Methodology

To define transfer levels, we first need a *source set*. The source set consisted of 20 problems and worked solutions, 5 from each of the problem types illustrated in Fig. 1. To generate learning curves for each of the 6 transfer levels, five *target training sets* were created for each of the transfer levels. Each target training set consisted of 4 quizzes, with one problem from each problem type in each quiz. Each problem represented a systematic transformation, based on the transfer level, from a problem in the source set. For example, to create the target training sets for transfer level 1, ETS generated 80 problems (i.e., 20 quizzes of four problems each). Each problem was a parameterization transfer to a problem in the source set. The Companion was given each target training set as a sequence of quizzes. After a quiz was administered, the Companion was given the worked solutions for the problems on that quiz. Thus, the worked solutions from earlier quizzes were available for use in solving later quizzes within the target training set. After each target training set, the Companion's memory was reset. Learning curves were created by computing the percentage of problems solved correctly on each quiz. That is, the first data point is the average of the results from the first quiz of each of the five target training sets, and the final data point is the average of the results from the fourth quiz of each target training set. The Companion was administered each target training set twice, one for each experimental condition. First, the Companion is given each target training set without access to the problems and worked solutions of the source set (the *no-transfer* condition). Then, to measure transfer learning, the Companion is given each target training set with access to the source set's problems and worked solutions (the *transfer* condition). Comparing the learning curves for these two conditions provides a measure of how much was learned via transfer from the source set.

There are three ways for transfer to manifest itself. (1) The system may get a jump start, i.e., the learning curve in the transfer condition has a higher y-intercept. (2) The system may learn faster (i.e., the system may reach asymptotic performance faster). (3) The system may reach a higher level of asymptotic performance. These are not mutually exclusive. Given the direct application of examples in analogical model formulation, we expect rapid learning; consequently, we focus on the jump start transfer results.

6.1.2. Results

Fig. 10 shows the learning curves¹⁰ for both the transfer and no-transfer conditions for each transfer level. TL-1, TL-3, TL-4, TL-5, and TL-6 all had 80 problems. TL-2 only had 40 problems as it was impossible to change the numeric parameters in such a way to qualitatively change the outcome for two Problem Types, 2 and 4. Recall that, in order to more precisely measure the effectiveness of analogical learning, we ensured that the Companion contained no equations of physics nor any other modeling knowledge about the domain. This means whenever a problem is solved, it is solved by analogical model formulation. Given that analogical model formulation requires remembered worked solutions to solve problems, the no-transfer condition always begins at 0% and improves as the Companion accumulates worked solutions. Using a randomization ANOVA measure [40], all transfer levels showed a statistically significant jump start ($p < .01$). For TL-1, TL-4, and TL-5, the jump start was 88%. Other levels were not as high: TL-2 was 50%, TL-3 was 25%, and TL-6 was 44%. This provides an average of 63.8% in jump-start performance, supporting our hypothesis that analogical model formulation can be used to solve AP Physics style problems, including handling these kinds of transfer.

6.2. Post-analysis and system modifications

While the jump start results support our hypothesis, it is important to understand whether the performance failures are due to analogical model formulation, or due to some other factor. Theoretically, there are many ways analogical model

¹⁰ Error bars for 0 standard error have been omitted for readability.

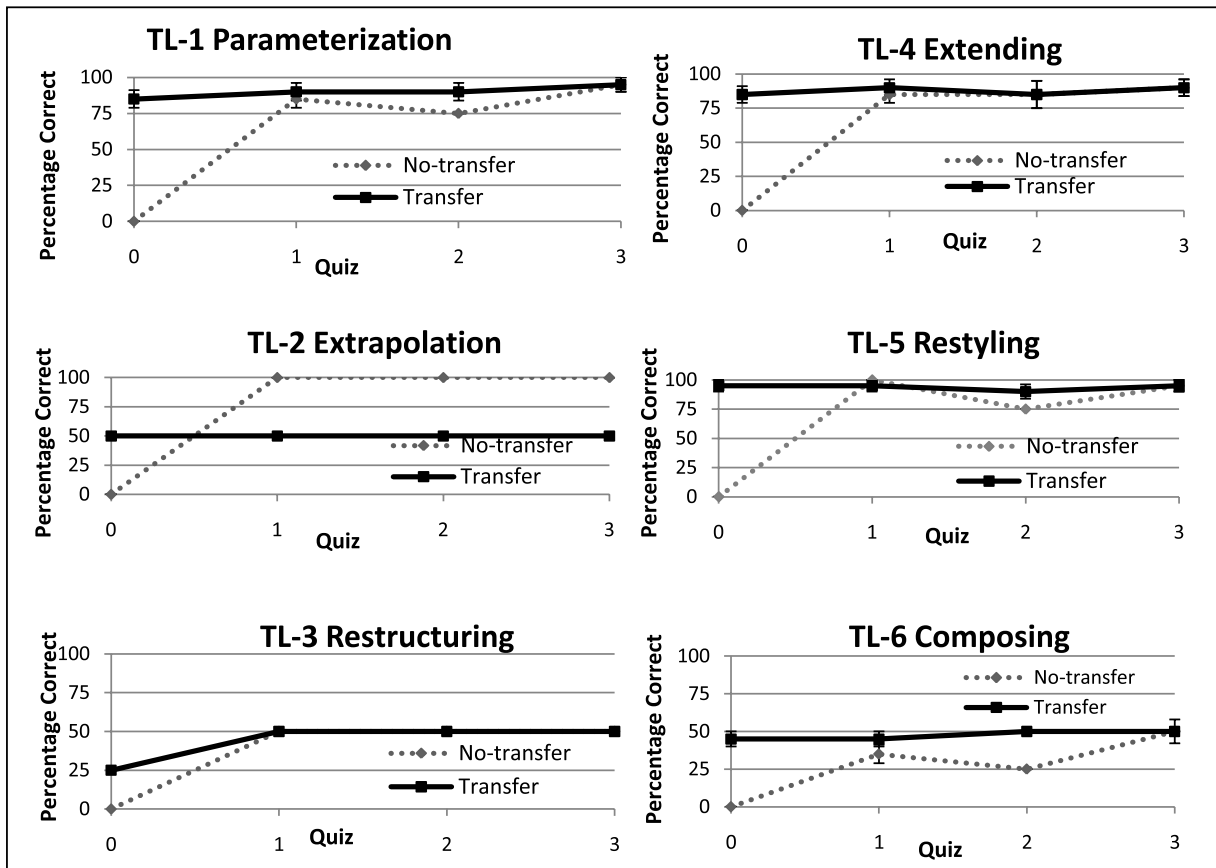


Fig. 10. Experiment 1 results, administered by the Educational Testing Service.

formulation can fail: The system can fail to find a precedent when one exists, mappings could be incorrect, or candidate inferences could be incorrectly analyzed. To our surprise, these problems accounted for only a small minority of the problem-solving failures. As illustrated below, the vast majority of the failures were due to human error in representing the problems and implementing the system.

In TL-2 (extrapolation), there is negative transfer, in that the no-transfer condition outperformed the transfer condition in later quizzes. This occurred because the Companion was repeatedly getting a problem correct for the wrong reasons in the no-transfer condition. An error in the worked solution representations for the target training set of Problem Type 3 caused the Companion to incorrectly assume a value for acceleration, which coincidentally led to the correct answer. The failure of the system to detect this and other recurring problems is leading us to focus effort on improving the Executive, as described in Section 8.

The low ceilings in the transfer condition in TL-2, and in both conditions in TL-3 (restructuring) and TL-6 (composing), are due to a combination of three limitations in the fixed components of the Companion's problem-solving strategies and a number of representation errors. The problem-solving strategy limitations were

1. The internal resource limit (i.e., maximum number of and/or graph nodes) was about 5% too low for some of the composing problems.
2. The algebra system was unable to correctly handle all of the necessary algebraic manipulation and equation comparison (e.g. trigonometry and composing symbolic and numeric problems).
3. The strategy of trying each value in turn for state elaboration problems, which make up 25% of TL-3, was very inefficient.

None of these problems concerns analogical model formulation, and we fixed them in the following manner. To solve the first problem, we increased the internal resource limit by 5%. Recall that learning modeling decisions is the focus of this work, not equation solving strategies. Therefore the second problem was solved by simply extending the algebra system to handle the necessary cases. The third problem was solved by developing a more efficient strategy for state elaboration problems. This involved altering the problem analysis and answer selection steps from our algorithm. Recall this example of a state elaboration problem:

Which of the following tensions is required to move a box of mass 8 kg from rest on the floor upward with constant acceleration when it is pulled vertically upward by a cord attached to the box? (a) 40 N; (b) 60 N; (c) 70 N; (d) 120 N.

During problem analysis, the new strategy identifies a *limit point quantity* whose value determines the consistency of a scenario. Here, the acceleration of the box is a limit point quantity. The scenario is consistent only if the acceleration is greater than 0 m/s^2 . Therefore, we use 0 m/s^2 as the limit point. We then use the query to determine the sought quantity, the tension of the cord. Next, the Companion assumes the limit point quantity value and proceeds with the algorithm to solve for the sought after quantity. After assuming an acceleration of 0 m/s^2 for the box, the Companion uses analogical model formulation to solve for the tension of the cord, 80 N. Instead of solving for the acceleration four different times, once for each assumption case, the new strategy is considerably more efficient, solving for the tension only once.

To select a multiple choice answer, the new strategy uses the scenario model to determine the qualitative relationship between the limit point quantity and the sought quantity. In this case, the acceleration of the box is qualitatively proportional to the tension in the cord (i.e., if the acceleration of the box is increased then the tension of the cord is increased). Since the problem indicates a consistent solution involves a positive acceleration, the Companion selects the multiple choice answer that is greater than 80 N, in this case, choice (d), 120 N. This new strategy only applies to state elaboration problems, which make up 25% of TL-3, Restructuring.

The only change to the analogical model formulation portion of the algorithm was the addition of one rule for verifying modeling decisions. When inferring a numeric parameter in quantity value assumptions, the Companion verifies that the units of the assumed value are applicable to the quantity type. For instance, the Companion will not assume 5 m/s for the acceleration of an object. Enabling Companions to learn such verification rules is an important element of our future work.

A close examination of the entire set of problems and worked solutions revealed two systematic kinds of representation errors.

1. Facts were sometimes omitted from the problem representations. For example, in some of the original representations, the correct answer was not listed as one of the answer choices, or the direction of a pulling event was not mentioned.
2. The agreed-upon conventions for representing worked solution steps were not always employed. For example, in the TL-2 extrapolation worked solutions for Problem 3 from Fig. 1, the exerted force is not sufficient to lift the object off the ground. The second to last step of the worked solution is to compare the computed acceleration, which is negative, against the boundary condition of 0 m/s^2 . The result of this step is that the object has an acceleration of 0 m/s^2 . While this worked solution step is a boundary condition check (i.e., `SanityChecking-PhysicsProblemSolution`), the original worked included it as an assumed value step (i.e., `DeterminingValueFromContext`).

Recall that the problems and worked solutions were automatically generated from templates, so these template-level bugs led to errors in all instances of that problem type at that transfer level. We fixed these systematic errors to generate a correct corpus of problems and worked solutions.

6.3. Experiment 2

To test our explanations for the results of Experiment 1, we conducted another experiment on the same sets of problems after fixing the representation and implementation errors.

6.3.1. Method

We use the same the experimental procedure, problems and worked solutions as Experiment 1 with the follow changes. First, we conducted the experiment, instead of the Educational Testing Service, due to budget cuts. Second, we used the corrected corpus of problems and worked solutions. Third, we made the system changes described in Section 6.2.

As in Experiment 1, we are testing the hypothesis that analogical model formulation transfers modeling knowledge across these six transfer levels. As before, we collected learning curves for each of the six transfer levels, and our analysis focuses on the jump starts in the transfer condition.

6.3.2. Results of Experiment 2

Fig. 11 contains the learning curves for each of the six transfer levels.¹¹ Across all the transfer levels, the Companion achieved a 95.8% jump start due to the source problems and worked solutions. Tls 1–5 all exhibited perfect transfer. The Companion performed at ceiling (100%) given just the source set worked solutions. On TL-6, the system recorded a jump start of 75%. Once again, all of the jump starts are statistically significant ($p < .01$).

These results illustrate that analogy can be used to formulate models in the domain of AP Physics problem-solving over a broad range of problems. Our retrieval rates and mapping rates were both 100%. That is, MAC/FAC always selected appropriate analogous worked solution(s) if they were available and SME always created a useful set of correspondences and candidate inferences between the worked solutions and the problems. The only failures of transfer involved limitations

¹¹ All data points have 0 standard error.

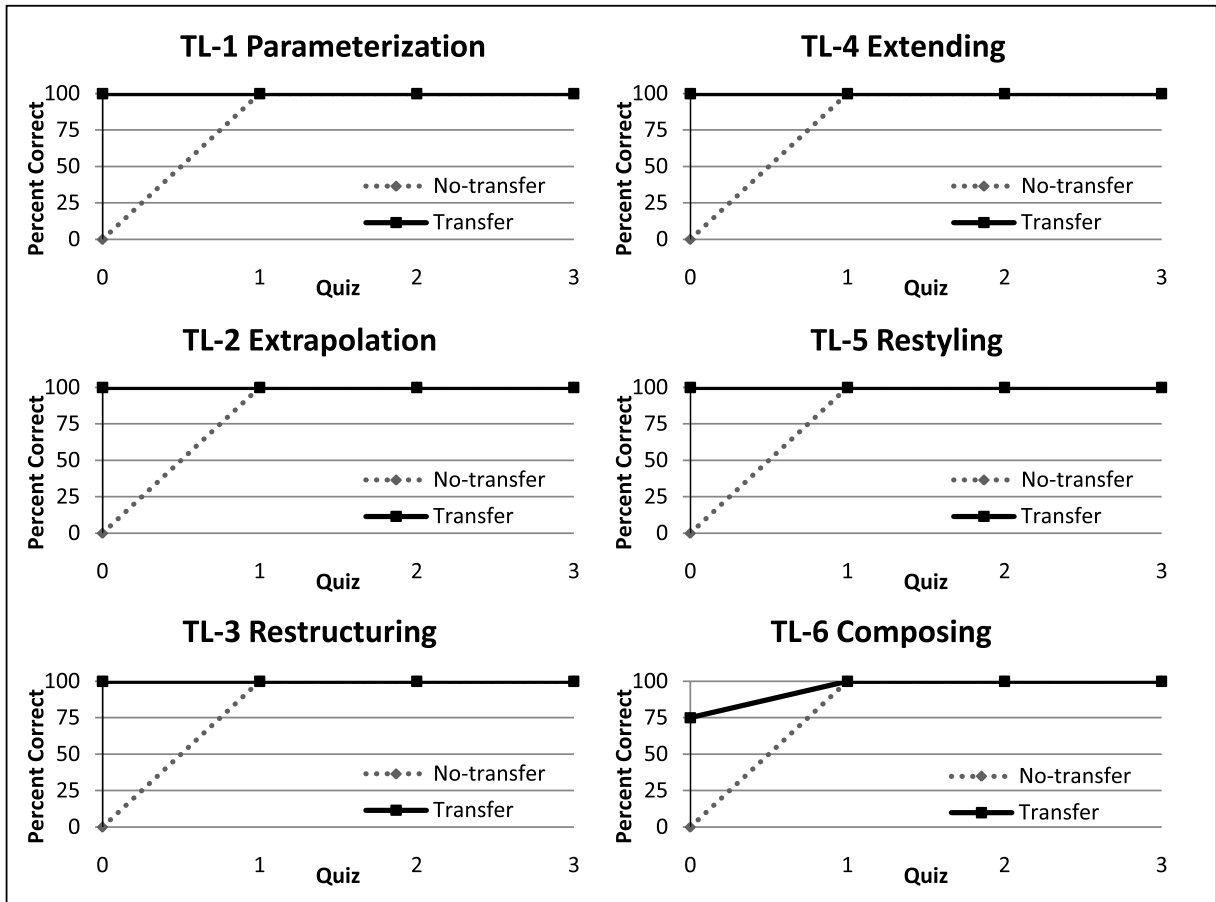


Fig. 11. Final evaluation results.

in our rules for verifying analogical inferences. In particular, our verification rules prevented a necessary modeling decision from being made on a set of difficult problems (25% of the composing problems). These composing problems involved source problems occurring on different planets and quantities referencing the planetary body of the problem explicitly (e.g., the work done by the gravitational force of an asteroid on a sliding block).

6.4. General discussion

These experiments support the claim that analogical model formulation is a robust method for transferring knowledge across these six transfer levels. First, the breadth of the materials and methods for evaluation are noteworthy. Drawn from four problem types, the 460 problems and worked solutions created for this evaluation included entities of 110 conceptual types and 144 unique relations. Second, in an experiment externally administered by ETS, a Companion achieved a significant jump start on all transfer levels (an improvement of 63.8% averaged across transfer levels) consisting of largely unseen problems. Finally, the second experiment provides a better understanding of the effectiveness of analogical model formulation across these six transfer levels. In this experiment, the Companion demonstrated perfect transfer across transfer levels 1–5. The 25% transfer failures on composing problems demonstrate a weakness of analogical model formulation. Specifically, while the hand-coded verification rules were effective across a broad range of scenarios, they do not handle all situations. The combination of these failures and the fact that we added the additional verification rule prior to Experiment 2 indicates that learning and refining these rules automatically is an important direction for future work.

Looking at analogical model formulation more generally, there are five potential failure modes:

- (1) The Companion does not have any examples analogous to the current scenario.
- (2) MAC/FAC fails to retrieve an analogous example, because the analogous examples are not similar enough to the current scenario.
- (3) The mapping resulting from SME may not result in candidate inferences necessary to make the model.
- (4) The verification rules may disallow a valid analogical modeling decision.
- (5) The verification rules may permit an incorrect analogical modeling decision.

To avoid the first type of failure, the system requires examples of all of types of models it needs to reason over. This knowledge engineering task is far easier for domain experts than the traditional method of providing complete and correct first principles knowledge bases. The second and third failure types are issues that arise from integrating analogical components into larger models, which is an important direction for analogy research [13]. The fourth and fifth failure types motivate our future research on learning reflective rules concerning evaluating analogical modeling decisions.

In these experiments, the analogical model formulation portion of the system failed in three ways. First, in the first trial of the no-transfer conditions, analogical model formulation failed because the system did not have any analogous examples in memory. Second, in the first experiment, analogical model formulation allowed some invalid analogical inferences for quantity values (e.g., permitting the assumption of an object's acceleration in invalid units, m/s). Third, in the composing problems of the second experiment, the verification rules disallowed valid analogical inferences. To reiterate, the retrieval (MAC/FAC) and mapping (SME) aspects of analogical model formulation worked perfectly, with no errors. Furthermore, a Companion using analogical model formulation successfully transferred knowledge across six systematic variations of relationships between AP Physics style problems.

7. Other related work

The history of AI contains numerous explorations of solving textbook problems. Within qualitative reasoning, de Kleer's [5] pioneering work in reasoning about sliding motion problems demonstrated that qualitative reasoning was required for solving many quantitative mechanics problems. The majority of the work on physics problem-solving has focused on equation solving. Two such systems, MECHO [3] and ISAAC [36], take natural language input and move to structural abstractions via collections of hand-coded rules to solve the problems. The connection between the handful of everyday entities each system knew about and the abstractions of physics were hand-coded, as was all of the domain knowledge. NEWTON, MECHO, and ISAAC were aimed at exploring how computers could solve physics problems at all, and the algebra system we use here is a direct descendant of ideas from Bundy's work. While there are other more capable algebra equation solvers available, they suffer from the same drawback as ours. That is, they are not extendable via learning. More recently, the HALO project [1] built knowledge-based systems that contained a few pages of hand-encoded textbook knowledge, to solve a small subset of AP Chemistry style problems. Like HALO, our system was evaluated on unseen problems in an externally administered evaluation. None of the early efforts nor HALO addressed learning, whereas learning domain knowledge from examples is the central focus of analogical model formulation.

Analogical problem-solving systems take a similar approach to ours, e.g. [48,34,39]. These systems solve new problems by transferring plans, rather than modeling knowledge, from previous problem-solving episodes. These systems require a complete and correct domain theory. Analogy is only used as a means of guiding the problem-solver. They could, with more effort, solve the problems without analogues. By contrast, the system described here does not require a complete domain theory. Moreover, it cannot solve anything without a prior example, making it an extreme test of analogical reasoning. Another difference is that the analogues for our system are worked examples, which are at a more abstract level than the problem-solver's internals, whereas the analogues for these prior systems were plans constructed by the problem-solvers themselves. It is unclear if plan-based analogical problem-solvers would do well on restructuring or composing problems. Restructuring problems require a different sequence of operations to solve for a new parameter; our method of only mining modeling information is agnostic with regard to the order in which information was used in worked solutions. Composing problems require combining concepts from multiple problems, which makes choosing plan steps more complex.

Of analogical problem-solving systems, Cascade [46] is perhaps most similar due to its focus on learning from examples and its domain of Newtonian physics. In addition to analogical search control knowledge, Cascade learns through resolving impasses while studying examples and solving problems. Cascade uses overly general rules to resolve an impasse and learn new domain knowledge. If the impasse occurs during example explanation and the overly general rules fail, Cascade allows this example to be used as a base for making analogical modeling decisions. In Cascade's evaluation, this use of analogy was employed specifically for one particularly difficult problem [47]. Instead of using analogy for modeling decisions as a last resort, our work demonstrates that analogy can play a primary role in model formulation. It is intriguing that analogical model formulation does as well as it does, and can even be used to explain a number of analogy event types found by VanLehn [45] in protocol studies [25]. It does seem likely that the best model of human reasoning (and most practical solution in engineering terms) is to both use analogical model formulation and learning that goes beyond accumulating examples.

In Case-Based Reasoning (CBR) systems, inferences are made about problems based upon previous cases. Most of today's CBR systems are based on feature-vectors, and hence lack the representational capacity to handle physics problems. In CBR systems which use relational representations (as we do), typically a heavy emphasis is placed on adaptation [29]. This frequently requires domain specific heuristics. In analogical model formulation, the adaptation is almost completely handled via structure mapping. Structure mapping theory uses common relational structure to constrain the inference process. The only domain specific heuristics for adaptation are the rules for evaluating the inferences and not for the matching process itself. This is done in two ways. (1) When numbers are included in inferences, we use the number from the worked solution, unless it is an assumed value inference with the base and target units in alignment. (2) The verification of analogical modeling decisions can result in the rejection of inferences. Like most CBR systems, our verification methods are currently hand-generated and consequently limited in their scope. Another important difference between CBR and analogical model

formulation is in regards to the how cases are retrieved. Retrieval systems for relational CBR tend to use indexing schemes that are carefully designed on a domain-specific and task-specific basis. By contrast, MAC/FAC is domain-independent and task-independent, and has been used to explain a number of psychological results [13].

Recent work in the cognitive architecture community emphasizes the importance of experiences in expanding the functionality of intelligent agents. Nuxoll [37] investigates a number of cognitive capabilities which an episodic memory should provide. Working within the SOAR cognitive architecture [31], Nuxoll demonstrates how episodic memory provides the following abilities: action modeling, retroactive learning, boosting other learning mechanisms, and virtual sensing. Our work differs by demonstrating how examples can be used to formulate models of new situations within a cognitive architecture.

Recently, there has been an increasing interest in transfer learning within the AI community. From a cognitive architecture perspective, ICARUS achieves near transfer through goal decomposition [4] and has been augmented with a representation mapping algorithm [43] to handle more distant types of transfer. We agree that finding domain mappings is critical to successful transfer. While the methods employed by ICARUS require abstracted domain theories in both the source and target tasks, analogical model formulation operates directly on the problem and a specific example without abstract modeling knowledge of the source or target task. Liu and Stone [32] use a version of SME to accelerate learning of state action policies in novel but similar tasks within the keep-away soccer domain. Instead of using structure mapping to accelerate learning, we use structure mapping as our learning mechanism. Hinrichs and Forbus [23] describe how analogy can be used to transfer learned qualitative models between scenarios in a turn based strategy game. Sharma et al. [44] use reinforcement learning for credit assignment and case-based reasoning to learn value functions for variations in a real-time strategy game. As in our work, these systems rely on similarities between the current situation and previous cases to drive knowledge transfer.

8. Conclusions

While Qualitative Reasoning (QR) research has made many important advances, the standard methodology has three core problems. First, most work assumes complete and correct domain theories. Such domain theories have proven hard to develop outside specialized engineering fields. Second, the problem of formulating models from everyday descriptions has essentially been ignored. This is due to several reasons, including a tendency to focus on technical domains (e.g., engineering, ecology, gene expression, weather prediction) and the perceived lack of off-the-shelf large-scale ontologies to facilitate such efforts. Third, most QR work does not take learning into account. This comes in large measure from viewing the enterprise as constructing expert reasoning engines, motivated by industrial applications.

The traditional method of constructing complete and correct domain theories by hand is applicable when the domain is narrow and the required reasoning is known in advance. On the other hand, broad domains, such as reasoning about physics scenarios, would greatly benefit from methods whose coverage incrementally extended. Coming from the perspective of cognitive architecture, our interest lies in such domains. In these cases, analogical model formulation is a promising approach. By focusing solely on direct application of examples, analogical model formulation provides a simple, but powerful, view of analogical reasoning.

A central hypothesis of the Companion cognitive architecture is that analogical reasoning and learning are central to human cognition. This places learning front-and-center, and leads directly to the approach of analogical model formulation. As the results from our experiments show, analogical model formulation can enable a system to learn to solve hard problems (i.e., AP Physics style problems) without a complete domain theory. Moreover, our experiments show that analogy provides a natural way to cope with the breadth of everyday knowledge in model formulation. We know of no other problem-solving experiments which demonstrate analogical learning over systematic variations of relationships between problems at this scale.

Clearly there is much work remaining to realize the full potential of analogical model formulation, and the potential for analogical learning in problem-solving more generally. Even in the realm of AP Physics, recall that this corpus of problems is drawn from roughly 20% of the Mechanics portion of the AP Physics exam, of which Mechanics is only one section. One goal is to expand the system to the point where it can learn all of the material covered by the AP Physics exam. As mentioned earlier, one limitation that must be overcome is relying on hand-coded rules for verifying analogical inferences and algebraic operations. There were ample signals to the system that something was wrong with its knowledge (i.e., multiple repeated failures in some of the transfer conditions in the first experiment), but the current version of the architecture was unable to exploit this information. Consequently, we are planning on improving the Executive functions in the Companion architecture, enabling the system to take responsibility for learning and refining its verification rules. By keeping track of successful and unsuccessful analogical modeling decisions, a Companion can learn in what context different types of modeling decisions are effective. Keeping and analyzing records of its problem-solving successes and failures should also provide the grist needed for formulating its own learning goals [41]. For example, consider the modeling decision of inferring a quantity value for the velocity of an object. By storing successes and failures, a Companion could be able to generalize that the analogical inferences for velocity values in meters per second would have a much higher success rate than other units. Also, as our experience in Experiment 1 indicates, creating error free predicate calculus representations by hand requires significant time and expertise. To avoid future representational encoding errors, additional work on Companions is focused on providing the user with natural interaction modalities (e.g., natural language and sketching), which will be used to

generate predicate calculus representations automatically [19]. In the future, we hope to provide the Companion with only the natural language description of the problem along with any accompanying diagrams.

In addition to testing the system on more problem types, learning verification rules, and providing users with more natural interaction modalities, we want to continue to extend Companions' ability to transfer knowledge. There are several additions that we believe to be essential to handle more distant transfer. First, we plan to move beyond learning by accumulating examples. One promising direction is to construct generalizations of the physical phenomena as *encapsulated histories* [12] using SEQL [30]. To utilize these learned abstract domain theories, the system would be required to make explicit modeling abstraction decisions. We have preliminary results from linear and rotational mechanics that SEQL generalization can be effective for learning to make modeling abstraction decisions [27]. Integrating these techniques to learn abstract domain theories would enable a Companion to transfer what it learns even more broadly.

Second, as a Companion accumulates generalizations in one area of physics, we will explore how dynamical analogies [38] can facilitate transfer learning into other areas of physics, such as electrical and hydraulic domains. Cognitive scientists have shown how cross-domain analogies are useful for learning new areas of physics [22,6]. To build an intelligent system that can achieve this ability, we believe two steps are important. First, we are developing a model of how cross-domain analogies can be used to build new domain theories [26]. Second, given the explanatory nature of cross-domain analogies, we plan on increasing interactivity to allow a user to work through complex cross domain analogies with a Companion. This would allow advice such as "heat flow is like water flow" to be understood and used by a Companion.

Acknowledgements

This research was supported by DARPA under the Transfer Learning program. We thank Patrick Kyllonen, Catherine Trapani and Vincent Weng at the Education Testing Service for help in experimental design, generating the testing materials, and administering the evaluation. We thank Cynthia Matuszek, Blake Shepard, and Casey McGinnis at Cycorp for helping work out the representation conventions for problems and worked solutions. We also thank Thomas Hinrichs, Jeff Usher, Praveen Paritosh, and Emmett Tomai for many interesting ideas and their work with us on building the Companion cognitive architecture.

Appendix A

Problem 2 representation

```
(isa Hyp-MT-ETS-Query-2-0-1 Microtheory)
(genlMt Hyp-MT-ETS-Query-2-0-1 PhysicsTestTakingAssumptionsMt)
(isa Movement-2-0-1 ProjectileMotion)
(isa Upward-Movement-2-0-1 ProjectileMotion)
(isa Astronaut-2-0-1 Astronaut)
(isa Ground-2-0-1 SurfaceRegion-Tangible)
(isa Planet-2-0-1 Planet)
(isa Throwing-2-0-1 ThrowingAnObject)
(isa BaseballBat-2-0-1 BaseballBat)
(except (ist PhysicsTestTakingAssumptionsMt (relationAllInstance eventOccursAt Event PlanetEarth)))
(groundOf Planet-2-0-1 Ground-2-0-1)
(performedBy Throwing-2-0-1 Astronaut-2-0-1)
(no-GenQuantRelnFrom in-ImmersedFully Planet-2-0-1 Atmosphere)
(eventOccursNear Throwing-2-0-1 Ground-2-0-1)
(objectThrown Throwing-2-0-1 BaseballBat-2-0-1)
(valueOf (MeasurementAtFn ((QPQuantityFn Speed) BaseballBat-2-0-1) (StartFn Upward-Movement-2-0-1))
(MetersPerSecond 6.5))
(eventOutcomes Throwing-2-0-1 Movement-2-0-1)
(primaryObjectMoving Movement-2-0-1 BaseballBat-2-0-1)
(firstSubEvents Movement-2-0-1 Upward-Movement-2-0-1)
(maximumMotionInDirection Movement-2-0-1 Upward-Movement-2-0-1 Up-Directly)
(primaryObjectMoving Upward-Movement-2-0-1 BaseballBat-2-0-1)
(directionOfTranslation-Throughout Upward-Movement-2-0-1 Up-Directly)
(valueOf (MeasurementAtFn ((QPQuantityFn Altitude) BaseballBat-2-0-1) (EndFn Upward-Movement-2-0-1))
(Meter 3))
(isa Acceleration-2-0-1 ScalarOrVectorInterval)
(isa NU-ETS-Query-2-0-1 KBContentTest-FullySpecified)
(isa NU-ETS-Query-2-0-1 ETSPhysicsSampleQuery)
(hypotheticalMicrotheoryOfTest NU-ETS-Query-2-0-1 Hyp-MT-ETS-Query-2-0-1)
(retainTerm (TestQueryFn NU-ETS-Query-2-0-1))
(testQuerySpecification NU-ETS-Query-2-0-1 (TestQueryFn NU-ETS-Query-2-0-1))
(querySentenceOfQuery (TestQueryFn NU-ETS-Query-2-0-1)
(valueOf ((QPQuantityFn AccelerationDueToGravity) Planet-2-0-1) Acceleration-2-0-1))
(termToSolveFor (TestQueryFn NU-ETS-Query-2-0-1)
(valueOf ((QPQuantityFn AccelerationDueToGravity) Planet-2-0-1) Acceleration-2-0-1) Acceleration-2-0-1)
(isa Hyp-MT-ETS-Query-2-0-1-WS Microtheory)
(genlMt Hyp-MT-ETS-Query-2-0-1-WS Hyp-MT-ETS-Query-2-0-1)
```

```
(multipleChoiceSingleOptionList NU-ETS-Query-2-0-1 (TheList (MetersPerSecondPerSecond 2.2) "A"))
(multipleChoiceSingleOptionList NU-ETS-Query-2-0-1 (TheList (MetersPerSecondPerSecond 0.5) "B"))
(multipleChoiceSingleOptionList NU-ETS-Query-2-0-1 (TheList (MetersPerSecondPerSecond 7.0) "C"))
(multipleChoiceSingleOptionList NU-ETS-Query-2-0-1 (TheList (MetersPerSecondPerSecond 19.5) "D"))
```

Worked solution for Problem 2 representation

```
(isa ETS-WorkedSolution-2-0-1 Individual)
(isa ETS-WorkedSolution-2-0-1 PhysicsWorkedSolution)
(comment ETS-WorkedSolution-2-0-1
  "An instance of PhysicsWorkedSolution. ETS-WorkedSolution-2-0-1 is a worked solution for the question posed
  by NU-ETS-Query-2-0-1.")
(workedSolutionForKBContentTest NU-ETS-Query-2-0-1 ETS-WorkedSolution-2-0-1)
(workedSolutionMtForTestMt Hyp-MT-ETS-Query-2-0-1 Hyp-MT-ETS-Query-2-0-1-WS)
(isa ETS-WorkedSolution-2-0-1-Step1 WorkedSolutionStep)
(isa ETS-WorkedSolution-2-0-1-Step2 WorkedSolutionStep)
(isa ETS-WorkedSolution-2-0-1-Step3 WorkedSolutionStep)
(isa ETS-WorkedSolution-2-0-1-Step4 WorkedSolutionStep)
(isa ETS-WorkedSolution-2-0-1-Step5 WorkedSolutionStep)
(isa ETS-WorkedSolution-2-0-1-Step6 WorkedSolutionStep)
(isa ETS-WorkedSolution-2-0-1-Step7 WorkedSolutionStep)
(isa ETS-WorkedSolution-2-0-1-Step8 WorkedSolutionStep)
(firstSolutionStep ETS-WorkedSolution-2-0-1 ETS-WorkedSolution-2-0-1-Step1)
(hasSolutionSteps ETS-WorkedSolution-2-0-1 ETS-WorkedSolution-2-0-1-Step2)
(hasSolutionSteps ETS-WorkedSolution-2-0-1 ETS-WorkedSolution-2-0-1-Step3)
(hasSolutionSteps ETS-WorkedSolution-2-0-1 ETS-WorkedSolution-2-0-1-Step4)
(hasSolutionSteps ETS-WorkedSolution-2-0-1 ETS-WorkedSolution-2-0-1-Step5)
(lastSolutionStep ETS-WorkedSolution-2-0-1 ETS-WorkedSolution-2-0-1-Step6)
(priorSolutionStep ETS-WorkedSolution-2-0-1-Step2 ETS-WorkedSolution-2-0-1-Step1)
(priorSolutionStep ETS-WorkedSolution-2-0-1-Step3 ETS-WorkedSolution-2-0-1-Step2)
(priorSolutionStep ETS-WorkedSolution-2-0-1-Step4 ETS-WorkedSolution-2-0-1-Step3)
(priorSolutionStep ETS-WorkedSolution-2-0-1-Step5 ETS-WorkedSolution-2-0-1-Step4)
(priorSolutionStep ETS-WorkedSolution-2-0-1-Step6 ETS-WorkedSolution-2-0-1-Step5)
(solutionStepOperationType ETS-WorkedSolution-2-0-1-Step1 CategorizingAPhysicsProblem)
(solutionStepUses ETS-WorkedSolution-2-0-1-Step1 (isa Throwing-2-0-1 ThrowingAnObject))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step1 (eventOccursNear Throwing-2-0-1 Ground-2-0-1))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step1 (groundOf Planet-2-0-1 Ground-2-0-1))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step1 (isa Planet-2-0-1 Planet))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step1
  (no-GenQuantReInFrom in-ImmersedFully Planet-2-0-1 Atmosphere))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step1 (objectThrown Throwing-2-0-1 BaseballBat-2-0-1))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step1 (eventOutcomes Throwing-2-0-1 Movement-2-0-1))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step1 (isa Movement-2-0-1 ProjectileMotion))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step1 (primaryObjectMoving Movement-2-0-1 BaseballBat-2-0-1))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step1 (firstSubEvents Movement-2-0-1 Upward-Movement-2-0-1))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step1 (isa Upward-Movement-2-0-1 ProjectileMotion))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step1
  (maximumMotionInDirection Movement-2-0-1 Upward-Movement-2-0-1 Up-Directly))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step1
  (primaryObjectMoving Upward-Movement-2-0-1 BaseballBat-2-0-1))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step1
  (directionOfTranslation-Throughout Upward-Movement-2-0-1 Up-Directly))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step1
  (valueOf (MeasurementAtFn ((QPQuantityFn Speed) BaseballBat-2-0-1) (StartFn Upward-Movement-2-0-1))
  (MetersPerSecond 6.5)))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step1
  (valueOf (MeasurementAtFn ((QPQuantityFn Altitude) BaseballBat-2-0-1) (EndFn Upward-Movement-2-0-1))
  (Meter 3)))
(solutionStepResult ETS-WorkedSolution-2-0-1-Step1
  (isa NU-ETS-Query-2-0-1 PhysicsProblem-DistanceVelocity))
(solutionStepResult ETS-WorkedSolution-2-0-1-Step1
  (isa NU-ETS-Query-2-0-1 PhysicsProblem-ConstantAcceleration))
(solutionStepOperationType ETS-WorkedSolution-2-0-1-Step2 SubstitutingBindingsForVariables)
(solutionStepUses ETS-WorkedSolution-2-0-1-Step2
  (isa NU-ETS-Query-2-0-1 PhysicsProblem-ConstantAcceleration))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step2
  (isa NU-ETS-Query-2-0-1 PhysicsProblem-DistanceVelocity))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step2
  (equationFormFor DistanceVelocityUnderConstantAcceleration
  (mathEquals (SquaredFn (MeasurementAtFn ((QPQuantityFn Speed) ?OBJ) (EndFn ?INTERVAL)))
  (PlusFn (SquaredFn (MeasurementAtFn ((QPQuantityFn Speed) ?OBJ) (StartFn ?INTERVAL)))
  (TimesFn 2 (MeasurementAtFn ((QPQuantityFn Acceleration) ?OBJ) ?INTERVAL))
  ((QPQuantityFn DistanceTravelled) ?OBJ ?INTERVAL))))))
```

```

(solutionStepUses ETS-WorkedSolution-2-0-1-Step2 (isa Upward-Movement-2-0-1 ProjectileMotion))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step2 (isa BaseballBat-2-0-1 BaseballBat))
(solutionStepResult ETS-WorkedSolution-2-0-1-Step2
  (equationForSolution ETS-WorkedSolution-2-0-1
    (mathEquals (MeasurementAtFn ((QPQuantityFn Acceleration) BaseballBat-2-0-1) Upward-Movement-2-0-1)
      (QuotientFn
        (DifferenceFn
          (SquaredFn (MeasurementAtFn ((QPQuantityFn Speed) BaseballBat-2-0-1)
            (EndFn Upward-Movement-2-0-1)))
          (SquaredFn (MeasurementAtFn ((QPQuantityFn Speed) BaseballBat-2-0-1)
            (StartFn Upward-Movement-2-0-1))))
          (TimesFn 2 ((QPQuantityFn DistanceTravelled) BaseballBat-2-0-1 Upward-Movement-2-0-1))))))
(solutionStepOperationType ETS-WorkedSolution-2-0-1-Step3
  DeterminingSpecificScalarOrVectorValuesFromContext)
(solutionStepUses ETS-WorkedSolution-2-0-1-Step3 (isa Throwing-2-0-1 ThrowingAnObject))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step3 (eventOccursNear Throwing-2-0-1 Ground-2-0-1))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step3 (groundOf Planet-2-0-1 Ground-2-0-1))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step3 (isa Planet-2-0-1 Planet))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step3
  (no-GenQuantRelnFrom in-ImmersedFully Planet-2-0-1 Atmosphere))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step3 (objectThrown Throwing-2-0-1 BaseballBat-2-0-1))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step3 (eventOutcomes Throwing-2-0-1 Movement-2-0-1))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step3 (isa Movement-2-0-1 ProjectileMotion))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step3 (primaryObjectMoving Movement-2-0-1 BaseballBat-2-0-1))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step3 (firstSubEvents Movement-2-0-1 Upward-Movement-2-0-1))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step3 (isa Upward-Movement-2-0-1 ProjectileMotion))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step3
  (maximumMotionInDirection Movement-2-0-1 Upward-Movement-2-0-1 Up-Directly))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step3
  (primaryObjectMoving Upward-Movement-2-0-1 BaseballBat-2-0-1))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step3
  (directionOfTranslation-Throughout Upward-Movement-2-0-1 Up-Directly))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step3
  (valueOf (MeasurementAtFn ((QPQuantityFn Altitude) BaseballBat-2-0-1)
    (EndFn Upward-Movement-2-0-1)
    (Meter 3)))
  (solutionStepUses ETS-WorkedSolution-2-0-1-Step3
    (valueOf (MeasurementAtFn ((QPQuantityFn Speed) BaseballBat-2-0-1) (StartFn Upward-Movement-2-0-1)
      (MetersPerSecond 6.5)))
    (solutionStepResult ETS-WorkedSolution-2-0-1-Step3
      (valueOf ((QPQuantityFn DistanceTravelled) BaseballBat-2-0-1 Upward-Movement-2-0-1) (Meter 3)))
    (solutionStepResult ETS-WorkedSolution-2-0-1-Step3
      (valueOf (MeasurementAtFn ((QPQuantityFn Speed) BaseballBat-2-0-1) (EndFn Upward-Movement-2-0-1)
        (MetersPerSecond 0)))
      (solutionStepResult ETS-WorkedSolution-2-0-1-Step3
        (valueOf (MeasurementAtFn ((QPQuantityFn Acceleration) BaseballBat-2-0-1) Upward-Movement-2-0-1)
          ((QPQuantityFn AccelerationDueToGravity) Planet-2-0-1)))
      (solutionStepOperationType ETS-WorkedSolution-2-0-1-Step4 SolvingAMathematicalEquation)
      (solutionStepUses ETS-WorkedSolution-2-0-1-Step4
        (equationForSolution ETS-WorkedSolution-2-0-1
          (mathEquals (MeasurementAtFn ((QPQuantityFn Acceleration) BaseballBat-2-0-1) Upward-Movement-2-0-1)
            (QuotientFn
              (DifferenceFn
                (SquaredFn (MeasurementAtFn ((QPQuantityFn Speed) BaseballBat-2-0-1)
                  (EndFn Upward-Movement-2-0-1)))
                (SquaredFn (MeasurementAtFn ((QPQuantityFn Speed) BaseballBat-2-0-1)
                  (StartFn Upward-Movement-2-0-1))))
                (TimesFn 2 ((QPQuantityFn DistanceTravelled) BaseballBat-2-0-1 Upward-Movement-2-0-1))))))
          (solutionStepUses ETS-WorkedSolution-2-0-1-Step4
            (valueOf ((QPQuantityFn DistanceTravelled) BaseballBat-2-0-1 Upward-Movement-2-0-1) (Meter 3)))
          (solutionStepUses ETS-WorkedSolution-2-0-1-Step4
            (valueOf (MeasurementAtFn ((QPQuantityFn Speed) BaseballBat-2-0-1) (StartFn Upward-Movement-2-0-1)
              (MetersPerSecond 6.5)))
            (solutionStepUses ETS-WorkedSolution-2-0-1-Step4
              (valueOf (MeasurementAtFn ((QPQuantityFn Speed) BaseballBat-2-0-1) (EndFn Upward-Movement-2-0-1)
                (MetersPerSecond 0)))
              (solutionStepUses ETS-WorkedSolution-2-0-1-Step4
                (valueOf (MeasurementAtFn ((QPQuantityFn Acceleration) BaseballBat-2-0-1) Upward-Movement-2-0-1)
                  ((QPQuantityFn AccelerationDueToGravity) Planet-2-0-1)))
              (solutionStepResult ETS-WorkedSolution-2-0-1-Step4
                (valueOf ((QPQuantityFn AccelerationDueToGravity) Planet-2-0-1)
                  (MetersPerSecondPerSecond (MinusFn 63.4))))
                (solutionStepOperationType ETS-WorkedSolution-2-0-1-Step5 SanityCheckingPhysicsProblemSolution)
                (solutionStepUses ETS-WorkedSolution-2-0-1-Step5 (isa Throwing-2-0-1 ThrowingAnObject))

```

```

(solutionStepUses ETS-WorkedSolution-2-0-1-Step5 (eventOccursNear Throwing-2-0-1 Ground-2-0-1))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step5 (groundOf Planet-2-0-1 Ground-2-0-1))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step5 (isa Planet-2-0-1 Planet))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step5
  (no-GenQuantReInFrom in-ImmersedFully Planet-2-0-1 Atmosphere))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step5 (objectThrown Throwing-2-0-1 BaseballBat-2-0-1))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step5 (eventOutcomes Throwing-2-0-1 Movement-2-0-1))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step5 (isa Movement-2-0-1 ProjectileMotion))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step5 (primaryObjectMoving Movement-2-0-1 BaseballBat-2-0-1))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step5 (firstSubEvents Movement-2-0-1 Upward-Movement-2-0-1))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step5 (isa Upward-Movement-2-0-1 ProjectileMotion))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step5
  (maximumMotionInDirection Movement-2-0-1 Upward-Movement-2-0-1 Up-Directly))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step5
  (primaryObjectMoving Upward-Movement-2-0-1 BaseballBat-2-0-1))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step5
  (directionOfTranslation-Throughout Upward-Movement-2-0-1 Up-Directly))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step5
  (valueOf (MeasurementAtFn ((QPQuantityFn Speed) BaseballBat-2-0-1) (StartFn Upward-Movement-2-0-1))
    (MetersPerSecond 6.5)))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step5
  (valueOf (MeasurementAtFn ((QPQuantityFn Altitude) BaseballBat-2-0-1) (EndFn Upward-Movement-2-0-1))
    (Meter 3)))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step5
  (valueOf ((QPQuantityFn AccelerationDueToGravity) Planet-2-0-1)
    (MetersPerSecondPerSecond (MinusFn 63.4))))
(solutionStepResult ETS-WorkedSolution-2-0-1-Step5
  (consistent
    (valueOf ((QPQuantityFn AccelerationDueToGravity) Planet-2-0-1)
      (MetersPerSecondPerSecond (MinusFn 63.4))))))
(solutionStepOperationType ETS-WorkedSolution-2-0-1-Step6 DeterminingTheBestAnswerFromASetOfChoices)
(solutionStepUses ETS-WorkedSolution-2-0-1-Step6
  (multipleChoiceSingleOptionList NU-ETS-Query-2-0-1 (TheList (MetersPerSecondPerSecond 2.2) "A")))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step6
  (multipleChoiceSingleOptionList NU-ETS-Query-2-0-1 (TheList (MetersPerSecondPerSecond 0.5) "B")))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step6
  (multipleChoiceSingleOptionList NU-ETS-Query-2-0-1 (TheList (MetersPerSecondPerSecond 7.0) "C")))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step6
  (multipleChoiceSingleOptionList NU-ETS-Query-2-0-1 (TheList (MetersPerSecondPerSecond 19.5) "D")))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step6
  (valueOf ((QPQuantityFn AccelerationDueToGravity) Planet-2-0-1)
    (MetersPerSecondPerSecond (MinusFn 63.4))))
(solutionStepUses ETS-WorkedSolution-2-0-1-Step6
  (consistent
    (valueOf ((QPQuantityFn AccelerationDueToGravity) Planet-2-0-1)
      (MetersPerSecondPerSecond (MinusFn 63.4))))))
(solutionStepResult ETS-WorkedSolution-2-0-1-Step6
  (testAnswers-SingleCorrectMultipleChoice NU-ETS-Query-2-0-1 "C"))

```

References

- [1] K. Barker, V. Chaudhri, Yi.S. Chaw, P. Clark, J. Fan, D. Israel, S. Mishra, B. Porter, P. Romero, D. Tecuci, P. Yeh, A question answering system for AP Chemistry: Assessing KR technologies, in: *Proceedings of the 9th International Conference on Knowledge Representation and Reasoning*, Whistler, BC, 2004.
- [2] C.K. Bennett, *Bennett Mechanical Comprehension Test*, The Psychological Corporation, San Antonio, TX, 1969.
- [3] A. Bundy, Solving mechanics problems using meta-level inference, in: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'79)*, Tokyo, Japan, 1979.
- [4] D. Choi, T. Konik, N. Nejati, C. Park, P. Langley, Structural transfer of cognitive skills, in: *Proceedings of the Eighth International Conference on Cognitive Modeling*, Ann Arbor, MI, 2007.
- [5] J. de Kleer, Multiple representations of knowledge in a mechanics problem-solver, in: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'77)*, Cambridge, MA, 1977.
- [6] B. Falkenhainer, An examination of the third stage in the analogy process: Verification-based analogical learning, in: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'87)*, Milan, Italy, 1987.
- [7] B. Falkenhainer, Analogical interpretation in context, in: *Proceedings of the Cognitive Science Society (CogSci-1990)*, Cambridge, MA, 1990.
- [8] B. Falkenhainer, Modeling without amnesia: Making experience-sanctioned approximations, in: *Proceedings of Qualitative Reasoning Workshop (QR'92)*, Edinburgh, UK, 1992.
- [9] B. Falkenhainer, K. Forbus, D. Gentner, The structure-mapping engine: Algorithm and examples, *Artificial Intelligence* 41 (1989) 1–63.
- [10] B. Falkenhainer, K. Forbus, Compositional modeling: Finding the right model for the job, *Artificial Intelligence* 51 (1991) 95–143.
- [11] T. Finin, D. McKay, R. Fritzson, R. McEntire, Kqml: An information and knowledge exchange protocol, in: Kazuhiro Fuchi, Toshio Yokoi (Eds.), *Knowledge Building and Knowledge Sharing*, Omaha and IOS Press, 1994.
- [12] K. Forbus, Qualitative process theory, *Artificial Intelligence* 24 (1984) 85–168.
- [13] K. Forbus, Exploring analogy in the large, in: D. Gentner, K. Holyoak, B. Kokinov (Eds.), *Analogy: Perspectives from Cognitive Science*, MIT Press, 2001.
- [14] K. Forbus, J. de Kleer, *Building Problem-Solvers*, MIT Press, Cambridge, MA, 1993.
- [15] K. Forbus, R. Ferguson, D. Gentner, Incremental structure-mapping, in: *Proceedings of the Cognitive Science Society (CogSci-1994)*, Atlanta, GA, 1994.

- [16] K. Forbus, D. Gentner, K. Law, MAC/FAC: A model of similarity-based retrieval, *Cognitive Science* 19 (1995) 141–205.
- [17] K. Forbus, D. Gentner, Qualitative mental models: Simulations or memories?, in: *Proceedings of the Eleventh International Workshop on Qualitative Reasoning (QR'97)*, Cortona, Italy, 1997.
- [18] K. Forbus, T. Hinrichs, Companion cognitive systems: A step towards human-level AI, *AI Magazine* 27 (2) (2006) 83–95.
- [19] K. Forbus, M. Klenk, T. Hinrichs, Companion cognitive systems: Design goals and lessons learned so far, *IEEE Intelligent Systems* 24 (2009) 33–46, July/August.
- [20] K. Forbus, D. Oblinger, Making SME greedy and pragmatic, in: *Proceedings of the Cognitive Science Society (CogSci-1990)*, Cambridge, MA, 1990.
- [21] D. Gentner, Structure-mapping: A theoretical framework for analogy, *Cognitive Science* 7 (2) (1983) 155–170.
- [22] D. Gentner, D.R. Gentner, Flowing waters or teeming crowds: Mental models of electricity, in: D. Gentner, A.L. Stevens (Eds.), *Mental Models*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1983.
- [23] T. Hinrichs, K. Forbus, Analogical learning in a turn-based strategy game, in: *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, Hyderabad, India, 2007.
- [24] M. Klenk, K. Forbus, Measuring the level of transfer learning by an AP Physics problem-solver, in: *Proceedings of AAAI-07: 22nd National Conference on Artificial Intelligence*, Vancouver, BC, 2007.
- [25] M. Klenk, K. Forbus, Cognitive modeling of analogy events in physics problem-solving from examples, in: *Proceedings of the 29th Annual Conference of the Cognitive Science Society (CogSci'07)*, Nashville, 2007.
- [26] M. Klenk, K. Forbus, Persistent mappings in cross-domain analogical learning of physics domains, in: *Proceedings of the 2nd International Analogy Conference*, Sofia, Bulgaria, 2009.
- [27] M. Klenk, S. Friedman, K. Forbus, Learning modeling abstractions via generalization, in: *22nd International Workshop on Qualitative Reasoning*, Boulder, CO, 2008.
- [28] M. Klenk, K. Forbus, E. Tomai, H. Kim, B. Kycckelhahn, Solving everyday physical reasoning problems by analogy using sketches, in: *Proceedings of AAAI'05*, Pittsburgh, PA, 2005.
- [29] J. Kolodner, *Case-Based Reasoning*, Morgan Kaufmann Publishers, San Mateo, CA, 1993.
- [30] S. Kuehne, K. Forbus, D. Gentner, B. Quinn, SEQL: Category learning as incremental abstraction using structure mapping, in: *Proceedings of the Cognitive Science Society (CogSci-2000)*, Philadelphia, PA, 2000.
- [31] J. Laird, A. Newell, P. Rosenbloom, SOAR: An architecture for general intelligence, *Artificial Intelligence* 33 (1987) 1–64.
- [32] Y. Liu, P. Stone, Value-function-based transfer for reinforcement learning using structure mapping, in: *Proceedings of AAAI'06: 21st National Conference on Artificial Intelligence*, Boston, MA.
- [33] C. Matuszek, J. Cabral, M. Witbrock, J. DeOliveria, An introduction to the syntax and content of cyc, in: *Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to the Knowledge Representation and Question Answering*, Stanford, CA, 2006.
- [34] J. Melis, E. Whittle, Analogy in inductive theorem proving, *J. Automat. Reason.* 22 (1999) 117–147.
- [35] P. Nayak, Causal approximations, *Artificial Intelligence* 70 (1994) 277–334.
- [36] G.S. Novak, Representation of knowledge in a program for solving physics problems, in: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'77)*, Cambridge, MA, 1977.
- [37] A. Nuxoll, *Enhancing intelligent agents with episodic memory*, PhD thesis, The University of Michigan, 2007.
- [38] H. Olson, *Solutions of Engineering Problems by Dynamical Analogies*, Van Nostrand, 1966.
- [39] T. Ouyang, K. Forbus, Strategy variations in analogical problem-solving, in: *Proceedings of AAAI'06: 21st National Conference on Artificial Intelligence*, Boston, MA, 2006.
- [40] J. Piater, P. Cohen, X. Zhang, M. Atighetchi, A randomized ANOVA procedure for comparing performance curves, in: *Proceedings of the Fifteenth International Conference Machine Learning*, Madison, WI, 1998.
- [41] A. Ram, D. Leake (Eds.), *Goal-Driven Learning*, MIT Press/Bradford Books, Cambridge, MA, 1995.
- [42] J. Rickel, B. Porter, Automated modeling for answering prediction questions: Selecting the time scale and system boundary, in: *Proceedings of the National Conference on Artificial Intelligence (AAAI'94)*, Seattle, WA, 1994.
- [43] D. Shapiro, T. Konik, P. O'Rourke, Achieving far transfer in an integrated cognitive architecture, in: *Proceedings of the Twenty-Second National Conference on Artificial Intelligence (AAAI'08)*, Chicago, IL, 2008.
- [44] M. Sharma, M. Holmes, A. Santamaria, J.C. Irani, C. Isbell, A. Ram, Transfer learning in real-time strategy games using hybrid CBR/RL, in: *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, Hyderabad, India, 2007.
- [45] K. VanLehn, Analogy events: How examples are used during problem solving, *Cognitive Science* 22 (3) (1998) 347–388.
- [46] K. VanLehn, Rule-learning events in the acquisition of a complex skill: An evaluation of Cascade, *J. Learning Sciences* 8 (1) (1999) 71–125.
- [47] K. VanLehn, R.M. Jones, M.T.H. Chi, A model of the self-explanation effect, *J. Learning Sciences* 2 (1992) 1–62.
- [48] M. Veloso, J. Carbonell, Derivational analogy in PRODIGY: Automating case acquisition, storage, and utilization, *Machine Learning* 10 (1993) 249–278.