PERSISTENT MAPPINGS IN CROSS-DOMAIN ANALOGICAL LEARNING OF PHYSICS DOMAINS

Matthew Klenk (matthew.klenk.ctr@nrl.navy.mil)
Navy Center for Applied Research in Artificial Intelligence,
Naval Research Laboratory, Washington, DC 20375 USA

Ken Forbus (forbus@northwestern.edu)
Qualitative Reasoning Group, Northwestern University
2133 Sheridan Rd, Evanston, IL 60201 USA

ABSTRACT

Cross-domain analogies are a powerful method for learning new domains. This paper extends the Domain Transfer via Analogy (DTA) method with the idea of *persistent mappings*, correspondences between domains that are incrementally built up as a system gains experience with a new domain. We evaluate DTA plus persistent mappings by learning three domains (rotational mechanics, electricity, and heat) by analogy with linear mechanics, showing that persistent mappings improves performance.

Keywords: Cross-Domain Analogy; Analogical Learning

1 INTRODUCTION

Cross-domain analogies are an important aspect of human reasoning. They are used by scientists to produce paradigm shifts (Gentner *et al.* 1997; Holyoak & Thagard 1989), and by students in learning new domains (Gentner & Gentner 1983). Textbook authors routinely exploit them when presenting new concepts, introducing a useful base domain explicitly and giving hints about correspondences.

Domain Transfer via Analogy (DTA) is a method of learning via cross-domain analogies. It uses analogies between examples as evidence about what non-identical predicate matches will be productive, re-using those correspondences to map equation schemas and control knowledge from the base to the target, creating a new target domain theory (Klenk & Forbus in press). This paper extends DTA with the idea of *persistent mappings*. Learning a complex domain via analogy typically does not happen all at once. The learner's understanding of the analogy grows incrementally. Persistent mappings represent the accumulation of knowledge about how the domains map. They are used to constrain subsequent matches, so that a structurally consistent set of knowledge is imported. To demonstrate the utility of persistent mappings, we show that adding them to DTA enables it to better learn multiple domains (rotational motion, electricity, heat) by analogy with linear motion. Moreover, we show that providing advice concerning retrieval and mappings, as textbook authors commonly do, can lead to substantial performance increases.

We begin by discussing the models of analogical processes used by DTA and the domain representations we use. Next, we describe the DTA algorithm and implementation. Then we describe the experiment and its results. We conclude with a discussion of related and future work.

2 ANALOGICAL PROCESSING

We use Gentner's structure-mapping theory (1983), which views analogy as a structural alignment process between two structured representations (the *base* and *target*), finding the maximal structurally consistent match. A key constraint is *tiered identicality*, meaning that there is a strong preference for matching identical predicates. In cross-domain analogies, non-identical functions are allowed to mach by default and non-identical relations match through *minimal ascension* (Falkenhainer 1988), illustrated below.

The Structure-Mapping Engine (SME) (Falkenhainer *et al.* 1989) models analogical matching. Given a base and target, SME produces one or more *mappings*. Each mapping is represented by a set of *correspondences* between *entities* and *expressions* in the base and target. Mappings also include a *structural evaluation score*, indicating the overall goodness of the match, and *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. SME can also be given a set of *constraints* as an input, consisting of required or excluded correspondences, which must be respected in the mappings it builds.

MAC/FAC (Forbus *et al.* 1995) models similarity-based retrieval. It is given a *probe* case and a case library as inputs, and optionally, constraints on correspondences. The first stage uses *content vectors*, automatically computed feature vectors where each predicate or relation of the case is represented by a dimension of the vector, whose strength is proportional to the number of statements using that predicate. The dot product of these vectors enables MAC to very rapidly select a few (at most three) candidates from the case library. The second stage uses SME to compare the structured representation of these candidates to the probe. The candidate match with the highest structural evaluation score is returned as the reminding.

Base Expression:
```
(stepUses WS-4-1-Step-2
   (objectTranslating Acc-4-1 Car-4-1))
```
Target Expression:
```
(stepUses WS-8-5-Step-2
   (objectActedOn Acc-8-5 Rotor-8-5))
```

Figure 1: Minimal ascension permits objectTranslating ↔ objectActedOn

Different domains are often represented using different predicates, especially when they are first being learned and underlying commonalities with previous knowledge have yet to be found. Minimal ascension allows non-identical predicates to match if they are part of a larger aligned structure and share a close common ancestor in the taxonomic hierarchy. Figure 1 shows two expressions that SME attempts to match because they have identical predicates, `stepUses`. In order to be included in the mapping, `objectTranslating` would have to correspond with `objectActedOn`. Minimal ascension allows this mapping because both relationships are descendants of `preActors` in the ResearchCyc[1] ontology, the taxonomic hierarchy used in this work.

3 REPRESENTING PHYSICS

We use the contents of the ResearchCyc knowledge base, plus our own extensions, to encode problems, worked solutions, and domain knowledge. Consequently, the vast majority of the relations and concepts we use were developed independently. For example, the everyday objects, relations, and events that appear in physics problems (e.g., "rotor", "car", and "driving") are already defined by the 30,000+ concepts and 8,000+ predicates in the KB. This reduces tailorability.

3.1 Problems and Worked Solution

The problem representations are direct translations from natural language into predicate calculus, without any abstraction or reasoning. Consider the problem, "A 75-V emf is induced in a .3 H coil by a current that rises uniformly from 0 to I in 2 ms. What is the value of I? (Problem 21-37, Giancoli 1991). This problem is represented by a case of 14 facts, defining the entities, events, known quantities and the sought parameter.

The worked solutions are represented at the level of explanations found in textbooks. They are neither deductive proofs nor problem-solving traces produced by our solver. Worked solutions consist of a sequence of steps. Each step has a *type*, *context*, and *results*. The ontology of step types was developed in collaboration with Cycorp in previous work (Klenk & Forbus 2007). The context consists of the facts which are antecedents for the step.

1 http://research.cyc.com/

The results indicate consequences of the step. In English, the worked solution for this problem is:

1. Categorize the problem as electrical
2. Definition of Inductance: V=L*di/dt
3. Solve for I = .5 amps

The formal representation for this solution consists of 30 facts. Worked solutions serve as examples from the domain. A central hypothesis of DTA is that structural similarities between examples can be used to create a domain mapping between two domain theories.

3.2 Domain Theories for Problem-solving

The domain theories consist of equation schemas and control knowledge. The equation schemas are encapsulated histories from QP theory (Forbus, 1984), as illustrated in Figure 2. They can be instantiated for any combination of entities that satisfy the type constraints and conditions. When that happens, the consequences are believed to hold. Control knowledge provides information about how to use schemas, e.g. that some schema should be considered *frame equations* (Pisan 1998) and thus are useful points for problem solving, or that one schema may be preferred over another.

```
(def-history DefOfInductance
 :participants
 ((theInductor :type Inductor-Idealized)
  (theEvent :type Conduction-Idealized))
 :conditions
   ((objectActedOn theEvent theInductor))
 :consequences
 ((equationFormFor DefOfInductance
   (mathEquals
    (AtFn ((QPQuantityFn VoltageAcross)
              theInductor
              DefOfInductance)
          theEvent)
    (TimesFn ((QPQuantityFn Inductance)
                 theInductor)
     (AtFn((QPQuantityFn RateOfCurrentChange)
              theInductor)
          theEvent))))))
```

Figure 2: Definition of self inductance encapsulated history

The textbook problems used here all ask for the values of specific quantities. During problem solving, the system attempts to find an equation with only the query quantity being unknown. If it finds such an equation, it solves it to determine the answer. Otherwise, it seeks relevant equations, guided by both what schema are available and the control knowledge. The algebra routines used are based on Forbus & De Kleer (1993).

4 HOW DTA + PERSISTENT MAPPINGS WORK

DTA assumes a known base domain consisting of equation schemas, control knowledge and problem/worked solution pairs. Given a problem that it cannot solve plus a worked solution in a new domain, it learns about the new domain using the following four steps (Figure 3): (1) learn the domain mapping, (2) initialize the target domain theory, (3) extend the target domain further via analogy, and (4) verify the new knowledge.

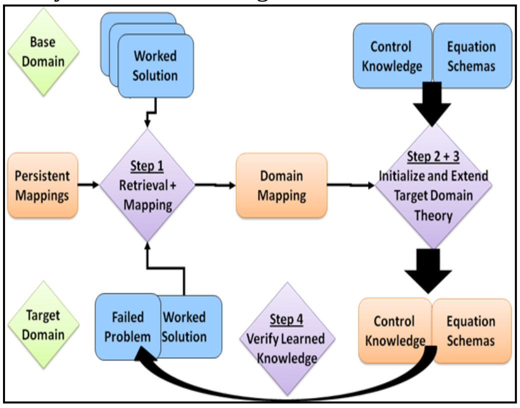

Figure 3: DTA model of cross-domain analogical learning

Adding persistent mappings changes this process by recording the domain mapping as part of what has been learned, so that when the next new problem in the target domain arrives, it uses those mappings as a starting point. We discuss each of these steps in turn next.

4.1 Step 1: Learn Domain Mapping

Different domains are typically represented with different predicates and conceptual types. Given a worked solution from the target domain, DTA finds an analogous worked solution from a domain it understands to start creating the mapping between two domains. This works because the language in which worked solutions are expressed, both in textbooks and our formal versions of them, use the same terms for the "connective tissue" of the explanation. That is, predicates like `stepUses` are generic across a wide span of domains.

Using the new target worked solution as the probe, MAC/FAC retrieves an analogous base worked solution. If there are already persistent mappings (i.e., this is not the first problem tackled in the target domain), these are provided to MAC/FAC as constraints. In the MAC stage, this makes the content vectors have more overlap than they would otherwise. During the FAC stage, the persistent mappings are treated as required correspondences. Each correspondence from the resulting mapping is added to the domain mapping when the base entity is mentioned in the base domain theory, as opposed to being something specific about that particular problem. Domain mappings include abstraction types (e.g., `PointMass` ↔ `Inductor-Idealized`), relations (e.g., `objectTranslating` ↔ `objectActedOn`) quantities (e.g., `ForceQuantity` ↔ `VoltageAcross`), and equation schemas (e.g. `DefOfNetForce` ↔ `DefOfInductance`).

4.2 Step 2: Initialize Target Domain

DTA uses the domain mapping to initialize the new domain theory. For each equation schema from the base domain mentioned in the domain mapping, DTA attempts to create a corresponding equation schema in the target domain. All of the equation schema's quantities and types must appear in the domain mapping for it to be transferrable. If they do, DTA performs the substitutions in the domain mapping on the base equation schema to derive a new target equation schema.

4.3 Step 3: Extend Target Domain

Next DTA performs another analogy, this time between the base and target domain theories themselves, again using the persistent mappings as required correspondences. To prevent non-analogous target items from interfering with the mapping, each equation schema in the target which does not participate in the domain mapping is prevented from mapping to any of the base equation schemas. DTA uses the candidate inferences from this analogy to infer more equation schemas and control knowledge for the target domain theory. For each equation schema, if all its participant types and quantities participate in the mapping, DTA performs the substitutions and adds the new equation schema to the target domain theory. Candidate inferences concerning control knowledge that refers to mapped equation schemas is also imported.

4.4 Step 4: Verify Learned Knowledge

While powerful, cross-domain analogies are risky and frequently contain invalid inferences. Therefore, DTA verifies the newly proposed knowledge by re-attempting the target problem. If this problem is solved correctly, DTA assumes that the learned knowledge is correct. Otherwise, DTA forgets the learned domain mapping, equation schemas, and control knowledge. Attempting to find other analogs is of course possible, but currently our system stops after the first failure.

After a successful transfer, the equation schemas and control knowledge are available for reasoning about future problems. The persistent mappings are also updated with new elements from the domain mapping, to improve the guidance provided to future analogies between that pair of domains.

4.5 Implementation on the Companions

We implemented DTA on the Companion Cognitive Architecture (Forbus *et al.* 2008). Processing in Companions is distributed, so that the domain reasoning is performed on one agent (the *Session Reasoner)* while retrieval is carried out on another (the *Retriever*). The experimental scripts used by the *Executive* agent provide a concise way of defining experiments and collecting data.

5 DYNAMICAL ANALOGIES

We created a corpus of four domains based upon Olsen's (1943) *Dynamical Analogies*: linear mechanics, rotational mechanics, electricity, and heat. Table 1 includes the analogous quantities from these domains.

Table 1: Dynamical analogy domains aligned by analogous quantities

Linear	Rotational	Electrical	Thermal
Force [F]	Torque [T]	Voltage across [V]	Temperature difference[T]
Speed [v]	Rate of rotation [ω]	Electrical current level [i]	Heat flow rate [q]
Linear deflection [x]	Rotational deflection [β]	Electrical charge [q]	Thermal energy [H]
Mass [F=ma]	Moment of inertia [T=Jα]	Inductance [V=Ldi/dt]	n/a
Linear momentum [p=mv]	Rotational momentum [p=J ω]	n/a	n/a
Linear kinetic energy [Ke=.5mv^2]	Rotational kinetic energy[Ke=.5 J ω^2]	Inductance energy[Energy=.5Li^2]	n/a
Linear compliance [F=x/C]	Rotational compliance [T= β/C]	Electrical capacitance [V=q/C]	Thermal capacitance [T=H/C]
Translational elastic potential [EPE=.5(x^2)/C]	Rotational elastic potential [EPE=.5(ω ^{2})/C]	Capacitance energy [Energy=.5(q^2)/C]	n/a
Linear damping [F=bv]	Rotational damping [T=Dω]	Electrical resistance [V=q/R]	Thermal resistance [q=T/R]
Power [P=Fv]	Power [P=Tω]	Power [P=Vi]	n/a

Previously, we demonstrated accelerated learning using DTA between linear and rotational kinematics (Klenk & Forbus in press). The dynamical analogy domains differ from the kinematics domains in several important dimensions. First, the new domains include superficially dissimilar domains such as mechanical and electrical systems. Second, they cover more phenomena than the kinematics scenarios, so a single cross-domain analogy between two worked solutions does not include all of the entities from the base and target domain theories. Third, the complexity of the problems is higher, requiring control knowledge to be included in the domain theories. Fourth, each of these domains has non-analogous elements (e.g., nothing corresponds to kinetic energy in thermal systems).

5.1 Advice-Taking and Iterative Cross-Domain Analogies

Textbooks and teachers present cross-domain analogies iteratively and provide students with advice, in the form of providing some correspondences between the domains. For example, Giancoli (1991) introduces rotational motion over an entire chapter, coming back to the analogy with linear motion repeatedly. During this presentation, the correspondences are sometimes made explicit (e.g. "Force is replaced by torque…Mass is replaced by moment of inertia…Linear acceleration is replaced by angular acceleration" (p. 197)).

We have already discussed how persistent mappings support iteration. Adapting DTA to take advice was done in two ways. First, a Companion can be instructed to invoke DTA with a given set of correspondences serving as additional persistent mappings, the equivalent of telling a student some of the correspondences. Second, a Companion can be instructed to invoke DTA with a given base worked solution, the equivalent of providing advice about the appropriate base problem.

6 EXPERIMENT

This experiment examines the following questions:

- Can DTA transfer knowledge across domains to solve novel problems?
- When retrieval fails, does providing the Companion with the analogous base solution lead to successful transfer?
- What are the effects of persistent mappings in learning domain mappings and aiding retrieval?

6.1 Materials

The problems were selected from a variety of physics resources, (Shearer *et al.* 1971; Giancoli 1991; Ogata 1997; Fogiel 1994; "Hooke's Law, Work and Elastic Potential Energy" 2009), with the following goals. First,

Table 2: DTA Results

Condition	Rotational Systems (7)		Electrical Systems (6)		Thermal Systems (2)		Total	
	Correct	Retrieval	Correct	Retrieval	Correct	Retrieval	Correct	Retrieval
DTA	2(29%)	3(43%)	1(17%)	2(33%)	0(0%)	0(0%)	3 (20%)	5 (33%)
DTA+B	6(86%)	n/a	4(67%)	n/a	2(100%)	n/a	12 (80%)	n/a
DTA+PM	3(43%)	3(43%)	1(17%)	3(50%)	0(0%)	0(0%)	4 (27%)	6 (40%)
DTA+PM+B	7(100%)	n/a	4(67%)	n/a	2(100%)	n/a	13 (87%)	n/a

the problem set includes each of the dynamical analogy quantities from Table 1. Second, the problems were limited in complexity to requiring at most three different physics equations. This is consistent with textbooks techniques of using more basic problems when introducing domains. Third, problems were favored which provided a worked solution, but worked solutions were created when necessary. Finally, since our algebra system does not yet handle calculus, problems were simplified to avoid it.

Predicate calculus representations for 22 problems and worked solutions were created: 7 from linear mechanics, 7 from rotational mechanics, 6 from electrical systems, 2 from thermal systems. The four domain theories together include 33 equation schemas and 56 quantities.

6.2 Method

The base domain for this experiment is linear mechanics, because it is the domain most students learn first. The Companion's case library contained the 7 linear mechanics problems and worked solutions. The base domain theory consisted of the equation schemas and control knowledge necessary to solve these 7 problems. The target domains are rotational, electrical, and thermal systems. The initial target domain theories include only the non-analogous equation schemas necessary to solve the problems (e.g. equations for computing the moment of inertia of a rotating point) which are impossible to learn via analogy. Notice that this is different than facts in the base which are not analogous to another domain – such facts should be ignored.

This experiment tests performance on a per problem basis in four independent conditions. First, the original DTA algorithm was run on each problem (the DTA condition). Second, each problem was presented with the analogous linear mechanics worked solution (DTA+B condition). The third and fourth conditions address the effects of persistent mappings. The DTA+PM condition provides each problem with the persistent mappings resulting from a successful run of DTA on the most similar problem. In the DTA+PM+B condition, both the persistent mappings and the correct retrieval were provided.

Each problem was scored as correct if the Companion was able to solve the target problem after the transfer. Retrievals were scored as correct if they found the closest analogous problem, and incorrect otherwise.

6.3 Results

Table 2 shows the results of the four conditions for each of the three transfer domains. All correctness results are statistically significant, since the odds of guessing a correct floating point value is essentially zero. Only retrieval for DTA+PM is statistically significant ($p < 0.02$), given only seven cases in the case library. In condition DTA, only 3 of the 15 target problems (20%) were solved correctly. This is because in that condition the correct analog was retrieved only 33% of the time: In DTA+B, the Companion solved 12 problems (80%). Condition DTA+PM shows that persistent mappings alone provide some value, retrieving 6 instead of 5 analogs correctly, and solving 4 instead of three problems. When both the relevant base and persistent mappings are provided (DTA+PM+B), the Companion does quite well, solving 13 problems (87%), showing that while having the correct analog provides most of the improvement, persistent mappings are still of value even in that case.

6.4 Discussion

Overall, the results show that DTA is able to transfer knowledge when it gets a correct retrieval, and that persistent mappings improve its performance. We examine the retrieval, transfer, and persistent mapping results in more detail next.

6.4.1 Retrieval

The primary cause for transfer failures was the inability to retrieve an analogous worked solution. This is consistent with psychological findings regarding the difficulties in the spontaneous retrieval of cross-domain analogies (Gick and Holyoak 1983). An inspection of the retrieval failures indicates that every failure occurred during the 1st stage of MAC/FAC. Given that the only overlap between most worked solution pairs between these domains are the "connective tissue" predicates of the worked solutions, there is little to discriminate on. Persistent mappings helped somewhat, by increasing the amount of predicate overlap, but this only mattered for one of the cases.

In this experiment, the Companion was only allowed one retrieval attempt. Allowing multiple retrievals could of course potentially improve performance, but we leave this possibility to future work.

6.4.2 Mapping and Transfer

Transfer depends upon the domain mapping learned from the analogy between worked solutions. Of the 41 problems in which DTA used the correct retrieval (adding across conditions), transfer was successful on 32 (78%) of them. The 9 worked solution mapping failures can be divided into three types: *merge failures*, *one-to-one failures*, and *incomplete mapping failures*. Merge failures occur when the mapping fails to include a necessary correspondence, because another correspondence already in the mapping blocks it. Recall that during SME, local match hypothesis are merged into structurally consistent global mappings. If two local matches include the same target element, only one can be included worked solution mapping. This can result in a failure to include important correspondences in the domain mapping.

One-to-one failures result from one-to-one constraint violations in the analogy between worked solutions. For example, the analogous equation schemas for kinetic energy ($Ke = .5mv^2$) and inductance energy ($E = .5Li^2$) each include three participants types. If the linear mechanics worked solution contains one entity for each of the participants, but the electrical worked solution includes only two entities, with one entity being used for multiple participants, then the resulting domain mapping will not include all the participant types. Consequently, the schema will not be transferred.

Incomplete mappings occur when the cross-domain analogy transfers only some of the equation schemas required to solve the target problem. This error occurred on one electrical problem, which requires equation schemas for both electrical capacitance, $V=q/C$, and capacitor energy, $Ce=.5(q^2)/C$. Although the cross-domain analogy successfully transfers the equation schema for electrical capacitance, the verification step fails, and the transferred knowledge and domain mapping are thrown out.

As the next section describes, persistent mappings allow DTA to overcome merge failures and incomplete mapping failures.

6.4.3 Persistent Mappings

The results from conditions DTA+PM and DTA+PM+B demonstrate that persistent mappings support incremental learning of a target domain theory through multiple cross-domain analogies. Furthermore, they improve cross-domain retrieval and mapping. In DTA+PM, persistent mappings enabled an additional correct retrieval. Persistent mappings prevent worked solution mapping failures. By already including useful correspondences between the domain theories, persistent mappings can avoid merge failures. By incorporating already learned aspects of the new domain, they are able to overcome incomplete mapping failures as well.

An underlying assumption of persistent mappings is that the entire cross-domain analogy satisfies the one-to-one constraint. That is, each element of the base domain theory corresponds to at most one element in the target domain theory. This assumption was not valid in this experiment, causing DTA to fail on the electrical power problem in the persistent mapping conditions even though it succeeded in previous conditions. Relaxing persistent mappings is thus an important direction for future work.

7 RELATED WORK

The major threads of related work concern Cognitive Science models of analogy and AI transfer learning

7.1 Cognitive Science Simulations

The closest system to ours is Falkenhainer's (1988) PHINEAS. PHINEAS used comparisons of (simulated) behavior to create an initial cross-domain mapping that was subsequently used to create a partial theory for the new domain. DTA differs by employing a more psychologically plausible retrieval mechanism and persistent mappings to incrementally construct complex cross-domain analogies.

Holyoak and Thagard's PI (1989) used a pragmatic theory of analogy to model solving a variation of the radiation problem through schema induction. PI only used analogy during problem-solving, and its retrieval model was never extensively tested. On the other hand, DTA makes analogies between examples as well as domain theories themselves, enabling the transfer of domain knowledge not explicitly referenced in the example (e.g., control knowledge in this experiment). Moreover, DTA tests its learned knowledge, and uses it to solve new problems from the target domain, whereas PI did neither.

Kühnberger *et al.* (2008) and Schwering *et al*. (2008) describe integrated architectures for achieving human-level intelligence in which analogy plays an important role. Like our work, these approaches emphasize ubiquity of analogical processing and the integration of analogy with other reasoning processes. A major point of departure between the above simulations and DTA is the scale of the tasks, which presents a major challenge for analogical learning systems (Forbus 2001).

7.2 Transfer Learning

Transfer learning is the improvement in performance on a new task through the transfer of knowledge from a related source task. Hinrichs & Forbus (2007) describe how analogy can be used to transfer learned qualitative models between scenarios in a turn based strategy game. As in DTA, examples are used to find the domain mapping between source and target domains. ICARUS has been augmented with a representation mapping algorithm to handle transfer tasks with different relations (Shapiro *et al.* 2008). While ICARUS's methods require abstract domain theories in both the source and target tasks, DTA can transfer abstract knowledge from the source domain with a single target example.

Lui and Stone (2006) use a version of SME to accelerate learning of state action policies in novel but similar tasks within the keep-away soccer domain. Taylor (2008) emphasizes the importance of the mapping between the states and actions of the source and target domains. Unlike these one shot transfers, DTA is an iterative process in which the target knowledge and domain mapping are incrementally verified and extended. For a more direct comparison, it is necessary to integrate DTA with domain learning techniques. This is an important direction for future research.

8 CONCLUSIONS

Using domain general methods of similarity-based retrieval and analogical matching, Domain Transfer via Analogy (DTA) enables the transfer of equation schemas and control knowledge between linear mechanical, rotational mechanical, electrical, and thermal domain theories. Persistent mappings support this process by building up a complex cross-domain analogy from successful local mappings. The results of our experiment, existing psychological research on memory retrieval, and quotations from common textbooks, all support our hypothesis that cross-domain analogical learning is an iterative process which in educational settings incorporates advice.

This paper suggests several future directions. First, self-modeling and task modeling is needed to decide when multiple retrievals are worth performing. Second, model-based diagnosis (de Kleer & Kurien 2003) could be used to debug analogies that go awry, allowing them to be fixed rather than discarded. Third, DTA experiments have focused on physics, but cross-domain analogical learning is useful in many other domains (e.g., strategy games and training simulators). Finally, since DTA is an iterative process which takes advice, integrating DTA in training simulators would enable AI agents to learn across a range of tasks and act as intelligent assistants providing feedback across scenarios and simulations.

9 ACKNOWLEDGMENTS

Matthew Klenk is supported by an NRC postdoctoral fellowship. This work was supported by the Cognitive Science Program of the Office of Naval Research.

10 REFERENCES

de Kleer, J. and Kurien, J. (2003) Fundamentals of model-based diagnosis. *SafeProcess*.

Falkenhainer, B. (1988). Learning from Physical Analogies. Technical Report No. UIUCDCS-R-88-1479, University of Illinios at Urbana-Champaign. (Ph.D. Thesis)

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

Fogiel, M., ed. (1994). *The Physics Problem Solver*. Research and Education Association.

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

Forbus, K. (2001). Exploring analogy in the large. In Gentner, D. *et al.* (Eds.) *Analogy: Perspectives from Cognitive Science*. MIT Press.

Forbus, K. and de Kleer, J. (1993). *Building Problem Solvers*, MIT Press.

Forbus, K., Gentner, D. and Law, K. (1995). MAC/FAC: A model of similarity-based retrieval. *Cognitive Science*, 19(2), 141-205.

Forbus, K., Klenk, M. and Hinrichs, T. (2008). Companion cognitive systems: Design goals and some lessons learned. In *the AAAI Fall Symposium on Naturally Inspired Artificial Intelligence*.

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

Gentner, D., Brem, S., Ferguson, R. W., Wolff, P., Markman, A. B., & Forbus, K. D. (1997). Analogy and creativity in the works of Johannes Kepler. In T. B. Ward *et al.* (Eds.), *Creative thought: An investigation of conceptual structures and processes*. American Psychological Association.

Gentner, D., & Gentner, D. R. (1983). Flowing waters or teeming crowds: Mental models of electricity. In *Mental Models*, by D. Gentner and A.L. Stevens. Eltham, London: Greenwich University Press.

Giancoli, D. (1991). *Physics: Principles with Applications*. 3rd Edition. Prentice Hall.

Gick, M. and Holyoak, K. (1983). Schema induction and analogical transfer, *Cognitive Psychology*. 15.

Hinrichs, T. & Forbus, K. (2007). Analogical learning in a turn-based strategy game. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*.

Holyoak, K. J., & Thagard, P. (1989). A computational model of analogical problem solving. In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning*. New York: Cambridge University Press.

"Hooke's Law, Work and Elastic Potential Energy" *Physics Homework Help from Experts*. 21 March 2009. <http://www.physics247.com/physics-homework-help/elastic-potential.php>.

Klenk, M. and Forbus, K. (2007). Measuring the level of transfer learning by an AP Physics problem-solver. In *Proceedings of the Association for the Advancement of Artificial Intelligence.*

Klenk, M. & Forbus, K. (in press). Domain Transfer via Cross-Domain Analogy.

Cognitive Systems Research DOI information: 10.1016/j.cogsys.2008.09.005

Kühnberger, K.-U., Geibel, P., Gust, H., Krumnack, U., Ovchinnikova, E., Schwering, A., and Wandmacher, T. (2008). Learning from Inconsistencies in an Integrated Cognitive Architecture. In *The Proceedings of the 1st Conference on Artificial General Intelligence.*

Lui, X., and Stone, P. (2006). Value-Function-Based transfer for reinforcement learning using structure mapping. *Proceedings of the 21st Association for the Advancement of Artificial Intelligence.*

Ogata, K. (1997). *System Dynamics*. Prentice Hall.

Olson, H. (1943). *Dynamical Analogies.* D. Van Nostrand.

Pisan, Y. (1998). *An integrated architecture for engineering problem solving*. Northwestern University: Department of Computer Science. (PhD Thesis)

Schwering, A.; Krumnack, U.; Kühnberger, K.-U., Gust, H. (2008). Analogy as integrating framework for human-level reasoning, In *The Proceedings of the 1st Conference on Artificial General Intelligence (AGI-08)*

Shapiro, D., Könik, T., and O'Rorke, P. (2008). Achieving far transfer in an integrated cognitive architecture. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*.

Shearer, J., Murphy, A., and Richardson, H. (1971). *Introduction to System Dynamics*. Reading, MA: Addison-Wesley

Taylor, M. (2008). *Autonomous Inter-Task Transfer in Reinforcement Learning Domains*. The University of Texas at Austin: Department of Computer Sciences (Ph.D. Thesis).