The Utility of Difference-Based Reasoning

Brian Falkenhainer
Qualitative Reasoning Group
Department of Computer Science
University of Illinois at Urbana-Champaign
1304 W. Springfield Avenue, Urbana, Illinois 61801

Abstract

The traditional approach to problem solving examines a current situation in isolation, ignoring the existence of previous experience. More recent analogical approaches look for previous, similar cases and attempt to infer further similarity from existing similarity. What has been overlooked is the power that identifying a *disanalogy* provides. Identifying disanalogies enables one to learn and reason by focusing on what is different between two similar situations, rather than on what is the same. This paper describes a technique called *difference-based reasoning* which exploits differences found between two otherwise identical situations to focus search and generate plausible hypotheses. The technique's power and diversity is demonstrated with implemented examples from theory formation, diagnosis, and failure explanation in planning.

1 Introduction

The predominate view of analogy in AI depicts a mechanism for the importation of knowledge from one domain or situation to another, based upon some form of underlying similarity between the two. What has been overlooked is the power that identifying a *disanalogy* provides. Identifying disanalogies enables one to learn and reason by focusing on what is different between two similar situations, rather than on what is the same. This paper describes a technique called *difference-based reasoning* which exploits differences found between two otherwise identical situations to focus search and generate hypotheses.

For example, consider the two situations illustrated in Figure 1. In case (a), the ball will never stop bouncing once set in motion (i.e., it stops when time reaches infinity). However, in case (b) the ball will stop bouncing in finite time. Why? Obviously, the answer must be related to the difference in angle of the incident wall, for the two situations are otherwise identical. The problem solving power achieved by detecting and focusing on this difference, and using it to ultimately arrive at an explanation, embodies the essence of difference-based reasoning.

This paper introduces difference-based problem solving and learning as an explicit technique and analyzes when it is applicable. Several examples from theory formation and revision, diagnosis, and failure explanation in planning demonstrate the technique's diversity. We conclude with a discussion of relevant work and issues for future research.

2 Difference-Based Reasoning

Difference-based reasoning (DBR) facilitates the resolution of expectation failure. Expectations may take on many forms, such as an expectation that two instances will behave the same (e.g., the pendulum example) or that a given instance will produce the desired consequence (e.g., planning, design, diagnosis, theory revision, etc.). It is *negative centered* in that it analyzes failure through the situation's differences with non-failing cases. This contrasts with the more traditional *positive centered* view of problem solving, which examines each situation in isolation, or using analogical methods, looks for previous *similar* cases and attempts to infer further similarity from existing similarity. Positive centered approaches ask questions like "How may this be solved?", "How was it solved before?", and "How did this fail before?". DBR capitalizes on questions of the form "How is the current case different from others that I've seen?" and "What did I change since the last time this worked?"

The key insight is that a significant amount of problem solving information may be obtained by analyzing differences between examples believed to be instances of the same concept, but which produce different results. By focusing on differences, we may quickly determine the characteristics relevant to the source of the problem. For example, if something doesn't work properly, people will often refer to a working example if one is available before resorting to first principles or cases of previous failures.

For a situation to be amenable to this technique, the following must be available:

- *Domain theory.* Sufficient axioms and vocabulary to draw meaningful conclusions.
- *Target example.* A description of the anomalous situation and the unexpected results it produces.

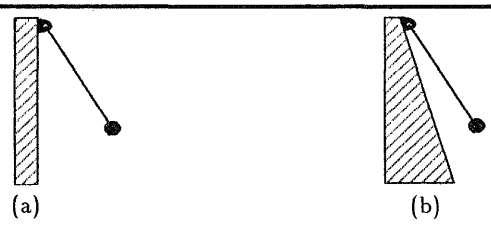

Figure 1: Two instances of the bouncing ball pendulum. In (a) the pendulum will never stop while in (b) the pendulum will stop in finite time. The impact of the pendulum's mass is assumed to be inelastic such that its kinetic energy is decreased in a fixed ratio at each impact.

- *Positive exemplar.* A description of an exemplar to compare against and the outcome it produces.

DBR guides an explanation of the anomalous result by first determining how the target instance differs from the positive exemplar. It then considers how these differences may account for their unequal behavior. Given a current example description and its unexpected outcome, DBR may be applied when an example or analogue of the expected behavior is available. The validity and utility of this procedure is based on the following theorem:

Difference theorem. Given sets of axioms F_p and F_t, in which G is derivable from F_p but is not derivable from F_t, the cause of failure of F_t must necessarily be traceable to axioms $\in \Delta$, where $\Delta = (F_p - F_t) \cup (F_t - F_p)$.

The proof follows easily from an assumption of monotonicity. The implication is that, when a positive exemplar is available, no explanation procedure should ever waste time exploring hypotheses that are not traceable to the difference set Δ. In many situations, this can be a very powerful guide. However, this is only the minimum that detecting differences provides. Depending upon the reasoning technique, domain, and instances involved, substantial information can also be obtained by examining the *manner* in which examples differ. Furthermore, the framework may be varied to create a constrained source of plausible conjecture rather than as part of a valid decision procedure.

2.1 Variations on the difference set

In general, Δ will be defined simply as the set of forms represented by $(F_p - F_t) \cup (F_t - F_p)$. However, there are some useful degrees of variability which should be considered.

- *Unidirectional.* In some problem settings, it may be sufficient to a-priori decide that the relevant information relates solely to explicit statements removed (or added) from the positive exemplar (i.e., the set $F_p - F_t$). However, these special cases must be considered in a domain dependent manner and are probably rare.
- *Preferential ordering.* In some circumstances, a decision procedure may be available to determine relevance or ordering within Δ. For example, in most domains it may be possible to ignore changes in color. The ability to focus on relevant differences is an important component of DBR and will be examined further in Section 2.3.
- *Qualitative analysis.* In physical domains, the set Δ may be reformulated to represent the net effect on each continuous quantity, rather than simply a collection of axioms unique to each instance. For example, Δ may answer qualitative questions about what quantities increased or decreased. We may then consider the instances as two states adjacent in time, and represent the net change from state S_p (positive exemplar) to state S_t (target example). This enables the use of standard qualitative reasoning techniques such as limit analysis (Forbus, 1984) and reduces the need for specific quantitative values.
- *Analogical comparisons.* In analogical settings, Δ is less exact in the same manner that identicality is less exact. Comparisons are more ones of similarity rather than of identicality, and the search for differences must reflect this. For analogical comparisons, we define the set Δ to be $[F_p - A_p(F_p, F_t)] \cup [F_t - A_t(F_p, F_t)]$, where $A_i(F_p, F_t)$ represents the elements from situation i found in the analogical mapping used by the performance engine. Thus Δ is simply the set of axioms from the two examples that were not placed in analogical correspondence. Note that this is incomplete in that it ignores the more subtle differences between items placed in analogical correspondence, an area where the analogical mapping itself may break down. This is a related, but unaddressed problem.

2.2 Identifying the positive exemplar

The utility of difference-based reasoning depends upon the availability of a useful exemplar and what the discernible differences are. First, there must be the expectation that the target example and the positive exemplar produce the same or analogous result. Second, the behavior of the two must differ in a detectable and usable manner. For example, when reasoning about physical systems, the behavioral difference must manifest itself in terms of observable quantities. Finally, the two instance descriptions must differ in a detectable and usable manner. For example, consider two circuit boards constructed from the same design specification, one operating properly and one not. While they must necessarily differ in structure at some level of granularity, this difference would typically not be detectable.

We claim that having a prior example and an expectation of *similar* function occurs in many cases of interest across diverse situations and domains. In some situations, the two contrasting examples will be explicitly presented, as in the statement of the pendulum problem. In analogical and case-based reasoning paradigms, the positive exemplar will already be present in the form of the base case or analogue. In many other situations, the positive exemplar must be accessed from a memory of previous instances or prototypes of the target concept. Note that this is typically much easier than the general analogical access problem, where one must retrieve from a potentially vast memory an example that is similar in some way to a current target example. In DBR, access is more a matter of looking up a prototypical example of the specific situation under investigation.

2.3 Using the Difference Set

There are several ways to focus on a relevant subset of differences or to use a given set of differences to guide problem solving. The simplest model would use the differences to focus forward chaining of rules (or in means-ends-analysis). This corresponds to the set-of-support strategy in resolution theorem proving. However, there are a number of standard, more sophisticated approaches available as well.

- *Domain knowledge and heuristics.* In many problem domains, knowledge is available to rule out irrelevant differences and enumerate the likely types of things to suspect. For example, in attempting to explain a malfunctioning mechanical subsystem, the analysis procedure should be able to distinguish between partially relevant and irrelevant structural differences. In a mechanical fault, the difference must be functionally related to the area of failure.

- *Establishing context.* Many reasoning systems are able selectively apply domain knowledge to a situation by identifying a current context during problem solving. If a difference is detected in one aspect of a situation, reasoning may be focused on that subarea.
- *Associative knowledge base.* Differences may be used as a source of knowledge to trigger remindings of relevant schemas, fault models, and problems associated with particular categories of change to a situation.
- *Interaction with other methods.* DBR is perhaps best viewed as an additional source of knowledge to be used in conjunction with existing techniques. For example, a diagnostic engine may query the difference set to see which candidate fault hypotheses are consistent with what is known. Alternatively, hypothesizing a particular fault model may create a query to look for a specific difference as confirming evidence, rather than a-priori obtaining all possible differences between two situations.

3 Issues and examples

Difference-based reasoning has two basic computational requirements. First, there must be a means to compare two situations and ascertain their difference. Second, an inference engine is required to apply domain knowledge to the explanation of failure and analysis of differences.

To satisfy the first requirement, the *Structure-Mapping Engine* (SME) (Falkenhainer, Forbus, & Gentner, 1986, 1987) is used to identify similarity or identicality. SME is a general tool for performing various types of analogical mappings. Given descriptions of two situations, SME identifies the best set of correspondences between them by analyzing their structural similarities. The difference set is then defined to be those aspects that failed to be placed in correspondence.

For the examples described in this section, two different inference engines were used. The first example uses Forbus' (1986) Qualitative Process Engine (QPE) to predict physical behaviors using models expressed in Qualitative Process theory (Forbus, 1984). The remaining two examples use a controllable, forward chaining rule system built atop an ATMS. In each example, we focus on the role of difference-based reasoning for aiding explanation of failure, and ignore the related issues of failure detection, repair, and storage.

DBR is intended for any failure explanation task that would benefit by having a working version to compare the failure against. Just how it is to be used can depend upon the reasoning task being considered. In the remainder of this section, we discuss examples from theory formation, diagnosis, and failure-driven learning in planning.

3.1 Theory formation and revision

Theory formation and revision attempts to develop and repair causal explanations of observed behavior. Within this framework, DBR may serve two purposes. First, it has the ability to focus hypothesis generation. When a theory's predictions are empirically contradicted, comparing how the anomalous situation differs from experiences consistent with the theory can help to identify the fringes of a theory's applicability and assign blame to its faulty elements. This is a standard component of traditional empirical learning techniques, but applied here in a knowledge-intensive framework.

In addition, analyzing differences provides a mechanism for base level conjectures when the domain theory is too weak or the observation too incomplete to attempt full explanation. If a behavior is seen to change with the introduction of a new relation, it would be reasonable to conjecture that the relation *caused* the change in behavior. The underlying mechanism may then be left for future theory generation.

3.1.1 Example: The bouncing pendulum

Consider the pendulum problem introduced in Section 1. The pendulum striking the vertical wall will never stop in finite time, while it will stop when striking the inclined wall. Many people approach this example with the expectation that both pendulums will behave the same.

To examine how DBR may assist in explaining this phenomenon, we have constructed a prototype implementation based upon Doyle's (1986) technique of layered approximation. The system's default model of the pendulum is highly abstract and simply predicts that both will oscillate forever (i.e., Zeno's paradox). Thus, the finite oscillation of the inclined case violates the system's expectation.

The system begins with structural descriptions of the two situations. Since the inclined pendulum violates the system's default model, it is given the inclined pendulum as the target example and the vertical pendulum as the positive exemplar. In the first stage of the analysis, SME is invoked to determine how the two situations differ. It responds with

Removed: `Equal-to(Contact-Theta(ball1), zero)`
Added: `Greater-than(Contact-Theta(ball1), zero)`

The *removed* relations are those present in the positive exemplar but missing in the target example, while the *added* relations are those uniquely part of the target example. The system's focus of attention is then aimed at the ball's contact angle by the rule "if an inequality between a quantity and zero changes, focus on that quantity". With this new focus of attention, the system reanalyzes the situation. This time, more detailed models are invoked which successfully predict different behaviors for the two pendulums. These predictions are shown in Figure 2.

In the simplest terms, recall that an oscillator will alternate about a central equilibrium point. In vertical case (a), the equilibrium point occurs at the exact point of contact, and thus half of the cycle will take place when the position is greater than the contact position. In inclined case (b), the zero force point is within the compression region, that is, the position of the ball is less than the contact position. This means that oscillation can take place without the ball ever leaving the surface of the wall and thus have no visible motion. In Figure 2(b), the cycles containing the two central paths correspond to no visible movement, with the oscillation taking place solely within

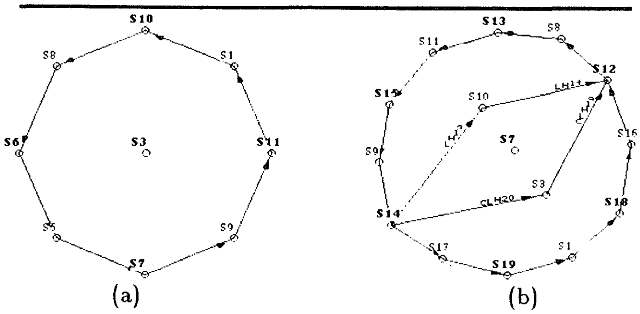

Figure 2: Alternate behaviors for the bouncing ball pendulum. In (a), there is no path to the stopped state. In (b), the cycles containing the two central paths correspond to no visible movement, with all of the oscillation occurring within the compressed region.

the compressed region.[1]

This example and its implementation has brought up a number of important issues that still must be resolved. First, the implemented example is overly simplistic and work on a more realistic version is in progress. Second, the type of flexible, dynamic reasoning about geometry needed for a more robust treatment is beyond current qualitative reasoning techniques, with or without the ability to focus on the problem's relevant characteristics.

3.2 Diagnosis

The literature on diagnosis centers around two techniques. The traditional approach is to store an explicit set of fault models for a device and attempt to determine which fault type applies to the current situation (see Davis, 1984 for a review). However, this requires all possible faults be anticipated in advance, and the number of applicable fault models may be extremely large. An alternate approach is to identify failed components by analyzing where the device's physical behavior deviates from the predictions of its corresponding model (e.g., Davis, 1984; DeKleer & Williams, 1987). However, model-based approaches assume a complete model, which is not always available. Each of these methods approach diagnosis by focusing solely on the current situation, where previous experience only appears in the form of fault models or probability distributions. For electrical circuits and many other complex domains, this is reasonable and about the best that one can expect. However, for domains readily characterizable by structural descriptions matching the granularity level of their faults, these methods fail to take advantage of all potentially available information.

Difference-based reasoning in the context of diagnosis reflects the common troubleshooting technique of consulting a working example for comparison. It recognizes that the structural flaw producing the deviant behavior may be identified by comparison to a correctly functioning example. For example, consider attempting to figure out why the driver's door on your car won't close all the way. It may be something blocking the hinges, bent hinges, something blocking the lock latch, a stuck lock latch, a bent door, bent lock latch, ice, etc. One method would be to enumerate all the possibilities and carefully examine the door for the existence of each. However, this is rather tedious and the number of possibilities is far too large. Furthermore, our model of the door is incomplete. While we have general knowledge of the door, we could not enumerate every piece and its relation to every other piece from memory. An alternative approach, using difference-based reasoning, would be to compare the door to one that is known to work properly – for example, the passenger door.

3.2.1 Example: The car door latch

In this section, we examine a simple difference-based diagnostic procedure capable of identifying and explaining why the car door fails to close.[2] We assume the existence of a full diagnostic system that is currently considering how the door latch might be at fault. A set of simplified domain rules for geometric reasoning are used. These rules represent specific knowledge about the latch mechanism, such as **qprop-(theta, wl)**, which expresses that the angle of the latch's displacement is inversely proportional to W_l (i.e., when W_l decreases, theta increases).

The DBR system is initially given a structural description of the driver's door latch and the task of explaining how it may prevent the door from closing. The system first retrieves a description of another latch that is known to work (i.e., the passenger door latch). These two latches are illustrated in Figure 3, with (a) being the suspected driver's door latch and (b) being the working passenger door version. After analyzing the two descriptions, SME finds that a piece of rubber is present in the working version, but missing in the driver's door latch. Furthermore, this piece of rubber is attached to the right end of the metallic part of the latch. The domain rules are then applied to analyze the net effect of adding the rubber piece. These rules enable the system to conclude that adding the piece of rubber increases the maximum angle achievable by the latch. Conversely, since the driver's door latch is missing the piece of rubber, its maximum angle is less than that for the working exemplar. The door's malfunction could be explained if the maximum angle for the driver's door mechanism had fallen below the threshold needed to latch.

The car door fault demonstrates several important benefits of difference-based diagnosis. First, the number of potential faults is extremely large, making a methodical analysis prohibitive. Even when focusing on the latch mechanism alone, there are many potentially relevant hypotheses. DBR's ability to focus the reasoning mechanism's attention on the relevant aspects makes the analysis tractable. Second, maintaining a set of prototypical examples can greatly facilitate isolating faults. Imagine the plausibility of a system proposing that a piece of rubber was missing if it had never encountered a working version! However, this must be carefully conditioned on

[1] This phenomenon, attributed to Meissner, is analyzed using Lienard's construction in (Stoker, 1950). Presented here is my personal qualitative explanation, which I arrived at by considering what effect the angle might have on the ball's behavior. Since I'm currently not certain of the explanation's accuracy, this example also demonstrates the potential for plausible hypothesis generation in difference-based reasoning and the usual problems that implies.

[2] The problem with the door is taken from actual experience.

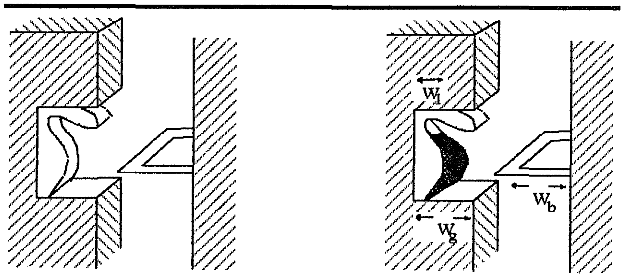

Figure 3: Car door latch. (a) The door only closes part way. (b) Properly functional door.

the type of domain under consideration. Structural differences must be easily recognizable and relevant to potential faults. Third, it is unreasonable to expect that a full model of the door is available for detailed analysis. Rather, having a prototype enables knowledge to be dynamically drawn from the exemplar on demand. Unfortunately, this example suffers from a need for geometric reasoning that AI systems still find quite difficult. Since the purpose is demonstration, the example has been greatly simplified. For example, the two dimensional geometry has been expressed in one-dimensional terms.

3.3 Failure explanation

Failure-driven learning recognizes that a useful way to schedule learning tasks is to wait until existing knowledge fails in some task (Hayes-Roth, 1983; Gupta, 1987; Hammond, 1987). Explaining this failure may then help prevent its reoccurrence in the future. Thus, it provides a reasonably focused and goal-driven method of learning. In this context, *failure* refers to a procedure that did not produce the desired outcome. The most straight forward approach is to resort to first-principles and heuristics in time of error (e.g., Hayes-Roth, 1983; Gupta, 1987). Unfortunately, given a sufficiently large set of interactions, explaining why a plan failed can become as difficult as the general diagnosis task. Hammond (1987) discusses the efficiency of accessing stored explanations of similar failures. However, the typical problems of associative memory access apply to retrieving similar failures. In addition, the number of previously explained failures and fault types grows with the breadth of the system's experiences, slowing failure explanation as more is learned.

In the context of explaining procedure failures, DBR seeks a previous success rather than a previous failure. It captures the notion that when a procedure fails, a useful heuristic is to focus on how the current situation differs from past examples of correct application. Locating previous failures has the drawback that the set of applicable failures is potentially large or potentially empty. Retrieving a successful instance should in general be easier than retrieving a negative instance. Having a successful instance simply means the procedure has worked in the past. However, the two approaches may still be used in concert, with differences from successful cases used to suggest previous failure explanations and guide their application to the current situation. Furthermore, there is potential for failure anticipation by constraining search into what possible interactions a new situation might cause.

3.3.1 Example: The flat soufle

Hammond (1987) describes an example in which his program CHEF constructs a plan to make strawberry soufle by adding strawberries to its existing vanilla soufle recipe. In this example, the plan fails because the soufle did not rise as expected. To explain the failure, CHEF analyzes the situation and the planning steps performed *in isolation*. It finds that the soufle fell because the 2.4 teaspoons of liquid and 60 teaspoons of whipped stuff represented an imbalance between the liquid and leavening in the recipe. There are several problems here. First, such detailed measurements are rarely available. Secondly, by examining the failed plan or previous similar failures, in isolation from the base case used to form the plan, valuable focusing information is lost. The only difference between the vanilla soufle and the strawberry soufle recipes is the strawberries.

In this section, a difference-based approach to explaining the strawberry soufle failure is described. The system was presented with descriptions of both cooking situations. This included the recipe used (e.g., 5 egg whites, etc.), the time and day, and the weather status. Furthermore, a statement of how the target strawberry soufle recipe failed was provided (i.e., `Texture(soufle, flat, S2)`). SME was used to analyze descriptions of the two recipes and determined that the time, day and weather had changed, and strawberries were added as a new ingredient.[3] After applying several rules, the system was able to conclude that the recipe's liquid content was increased due to the strawberries. Rather than having to know how much liquid and leavening is present, we may then compare the two situations and derive the relevant information with the following rule:

```
c-balance(q1, q2, s1) ∧ equal[V(q1, s1), V(q1, s2)]
                      ∧ greater[V(q2, s1), V(q2, s2)]
 ⇒  ¬c-balance(q1, q2, s2)
```

This rule states that if there is a chemical balance between two quantities in state S1, and only one of the quantities is greater in state S2, then there cannot be a chemical balance between them in S2. The lack of balance between the liquid and leavening quantities in the strawberry soufle explains why its texture was flat. An alternative approach would have combined the knowledge that too much liquid can leave a soufle flat with the knowledge that the added strawberries contain liquid.

4 Related Work

Oppenheimer (1956) stressed the importance of identifying disanalogies to find unifying abstractions and refine scientific theories developed from analogy. This is certainly important to scientific theory formation, and DBR stems from the development of PHINEAS, a program designed to investigate analogical theory formation and revision (Falkenhainer, 1987). DBR is a natural complement to analogical learning, which suffers from potential

[3] Given a complete case-based problem solver, the differences would be explicit in the transformations used to modify the original vanilla soufle recipe. In this example, we simply use SME to automatically provide the equivalent information.

inaccuracies that require refinement. However, the general difference-based reasoning mechanism goes far beyond abstraction formation and theory revision in applicability.

Weld's (1987) comparative analysis technique is a procedure for determining the effects of qualitative changes to a system's continuous parameters. It is not applicable to structural modifications, additions, or deletions. However, under these limiting conditions, it would be extremely useful for the second stage in DBR, where the effects of a given difference set are analyzed.

Identifying similarities and differences has its strongest machine learning roots in inductive, empirical methods. These methods form characteristic or discriminant descriptions for conceptual classification. Differences in this sense are features or relations that may be used to prevent overlap among elements of distinct conceptual categories. Among the empirical methods, DBR is most like Winston's (1975) *near miss* approach, which focuses on differences in example descriptions to hypothesize changes to a developing concept description (e.g., adding MUST-NOT-ABUT if the two supports touch in a negative example of an arch). However, these differences were never used in conjunction with domain knowledge for problem solving.

5 Discussion

Difference-based reasoning is applicable to a wide range of domains and reasoning tasks. This has been demonstrated by examples from theory formation, diagnosis, and planning failure explanation. It is relevant to situations in which an expectation was violated and an instance of the desired performance is available. It may be used in purely deductive settings to guide the explanation process or in inductive settings as a source of focused conjecture. When the domain vocabulary is too unconstrained to identify the problem from first principles, the technique may become an absolute necessity.

There are still many issues to be explored. For example, we have yet to examine specialized methods for analyzing a given set of differences. DBR has potential relevance for a much broader range of problems than currently examined. More examples from new problem domains are needed to better understand how it may be used.

The next phase of research will be to fully integrate the method with existing theory revision techniques in PHINEAS (Falkenhainer, 1987). This will provide a comprehensive framework to further investigate its role in large learning tasks, particularly how it can assist in repairing faulty analogies. For example, Falkenhainer & Rajamoney (1988) describe how an observation of liquid in a closed container violated their model's expectation that all contained liquids evaporate. They show how its revision may be facilitated by examining analogous behaviors, such as dissolving stopping due to saturation. However, that approach looked solely at the anomalous behavior and never noticed how the situation differed from the previous observation. Specifically, the only difference was the use of a closed container – an indication that one's models of finite capacity may be applicable.

6 Acknowledgements

The pendulum example, which gave me the initial idea of DBR, was first described to me by Ken Forbus in the context of interesting problems for qualitative reasoning. This work has benefited from insightful discussions with John Collins, Dennis Decoste, and Ken Forbus.

This research is supported by an IBM Graduate Fellowship and by the Office of Naval Research, Contract No. N00014-85-K-0559.

References

[1] Davis, R., Diagnostic reasoning based on structure and behavior, *Artificial Intelligence* **24**, 1984.

[2] DeKleer, J. and B. Williams, Diagnosing multiple faults, *Artificial Intelligence* **32**, 1987.

[3] Doyle, R.J., Constructing and refining causal explanations from an inconsistent domain theory. *Proceedings of AAAI-86*, Philadelphia, August, 1986.

[4] Falkenhainer, B., An examination of the third stage in the analogy process: Verification-Based Analogical Learning, *Proceedings of IJCAI-87*, August, 1987.

[5] Falkenhainer, B., K.D. Forbus, D. Gentner, The Structure-Mapping Engine, *AAAI-86*, 1986.

[6] Falkenhainer, B., K.D. Forbus, D. Gentner, The Structure-Mapping Engine: Algorithm and Examples, *Artificial Intelligence* (to appear). Also appears as Technical Report UIUCDCS-R-87-1361, Dept. of Computer Science, University of Illinois, July, 1987.

[7] Falkenhainer, B. and S. Rajamoney, The Interdependencies of Theory Formation, Revision, and Experimentation, *Proceedings of the Fifth International Machine Learning Conference*, Ann Arbor, June, 1988.

[8] Forbus, K.D., Qualitative Process Theory. *Artificial Intelligence* **24**, 1984.

[9] Forbus, K.D., The Qualitative Process Engine, Technical Report UIUCDCS-R-86-1288, Department of Computer Science, University of Illinois, 1986.

[10] Gupta, A., Explanation-Based Failure Recovery, *Proceedings of AAAI-87*, Seattle, July, 1987.

[11] Hammond, K.J., Explaining and Repairing Plans that Fail, *Proceedings of IJCAI-87*, August, 1987.

[12] Hayes-Roth, F., Using proofs and refutations to learn from experience. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell (Eds.) *Machine Learning: An artificial intelligence approach, Vol. I*, 1983.

[13] Oppenheimer, R., Analogy in science. *American Psychologist* **11**, 127-135, 1956.

[14] Stoker, J.J., *Nonlinear Vibrations*, Wiley Interscience, New York, 1950.

[15] Weld, D., Comparative Analysis. *Proceedings of IJCAI-87*, Milan, Italy, August, 1987.

[16] Winston, P.H., Learning structural descriptions from examples. In P.H. Winston (Ed.) *The psychology of computer vision*, McGraw-Hill, New York, 1975.