The Structure–Mapping Engine*

Brian Falkenhainer
Kenneth D. Forbus
Qualitative Reasoning Group

Dedre Gentner
Psychology Department

University of Illinois
1304 W. Springfield Ave
Urbana, Illinois, 61801

ABSTRACT

This paper describes the *Structure-Mapping Engine* (SME), a cognitive simulation program for studying human analogical processing. SME is based on Gentner's *Structure-Mapping theory* of analogy, and provides a "tool kit" for constructing matching algorithms consistent with this theory. This flexibility enhances cognitive simulation studies by simplifying experimentation. Furthermore, SME is very efficient, making it a candidate component for machine learning systems as well. We review the Structure-Mapping theory and describe the design of the engine. Next we demonstrate some examples of its operation. Finally, we discuss our plans for using SME in cognitive simulation studies.

1. INTRODUCTION

This paper describes the *Structure-Mapping Engine* (SME), a cognitive simulation program we have built to explore the computational aspects of Gentner's *Structure-Mapping theory* of analogical processing. SME is both flexible and efficient. It provides a "tool kit" for constructing matchers consistent with the kinds of comparisons sanctioned by Gentner's theory. A matcher is specified by a collection of rules. The rules can include strengths of evidence, and the program uses these weights and a novel procedure for combining the local matches constructed by the rules to efficiently produce and weigh all consistent global matches. The efficiency and flexibility of this matching algorithm suggests it would also be a viable component for machine-learning systems.

Cognitive simulation studies can offer important insights for understanding the human mind. Unfortunately, cognitive simulation programs tend to be complex and computationally expensive (c.f. [Anderson, 1983; Van Lehn, 1983]). Being complex makes the relationship between the program and the theory obscure. In addition, it is harder to make computational experiments and account for new data if the only way to change the program's operation is surgery on the code. Being computationally expensive means performing fewer experiments, and thus exploring fewer possibilities. There have been several important AI programs that study the computational aspects of analogy, but they were not designed to satisfy the above criteria (e.g. Burnstein, 1983; Winston, 1980, 1982).

The next section briefly reviews Gentner's Structure-Mapping theory. Section 3 describes SME's organization and its novel matching algorithm. Section 4 illustrates SME's operation on several examples, and Section 5 describes our plans for future development and for using it in psychological experimentation.

2. THE STRUCTURE–MAPPING THEORY

The theoretical framework for this research is the Structure-Mapping theory of analogy (Gentner, 1980, 1982, 1983; Gentner & Gentner, 1983). This theory describes the set of implicit rules by which people interpret analogy and similarity. The central intuition is that an analogy is a mapping of knowledge from one domain (the base) into another (the target) which conveys that a system of relations known to hold in the base also holds in the target. The target objects do not have to resemble their corresponding base objects. Objects are placed in correspondence by virtue of their like roles in the common relational structure.

Given collections of objects $\{b_i\}$, $\{t_i\}$ in the base and target representations, respectively, the tacit rules for constructing the analogical mapping M can be formalized as follows:** Objects in the base are placed in correspondence with objects in the target:

$$M: \quad b_i \text{ --> } t_i$$

Predicates are mapped from the base to the target according to the following mapping rules:

(1) Attributes of objects are dropped:
e.g. $\mathrm{RED}(b_i) \not\rightarrow \mathrm{RED}(t_i)$

(2) Relations between objects in the base tend to be mapped across:
e.g. $\mathrm{COLLIDE}(b_i, b_j) \text{ --> } \mathrm{COLLIDE}(t_i, t_j)$

(3) The particular relations mapped are determined by *systematicity*, as defined by the existence of higher-order*** constraining relations which can themselves be mapped:

e.g. $\mathrm{CAUSE}[\mathrm{PUSH}(b_i, b_j), \mathrm{COLLIDE}(b_j, b_k)] \text{ --> } \mathrm{CAUSE}[\mathrm{PUSH}(t_i, t_j), \mathrm{COLLIDE}(t_j, t_k)]$

For example, consider the analogy between heat-flow and water-flow. Figure 1 shows a water-flow situation and an analogous heat-flow situation. Figure 2 shows the representation a learner

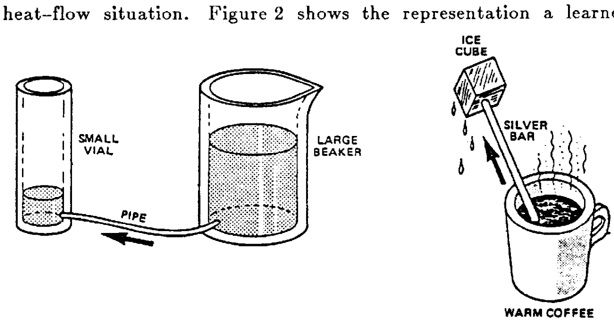


Figure 1. Two Physical Situations Involving Flow (adapted from Buckley, 1979, pp 15–25).

* This research is supported by the Office of Naval Research, Contract No. N00014-85-K-0559.

** Besides analogy, other kinds of similarity can be characterized by the distribution of relational and attributional predicates that are mapped. In *analogy*, only relational predicates are mapped. In *literal similarity*, both relational predicates and object-attributes are mapped. In *mere-appearance* matches, it is chiefly object-attributes that are mapped.

*** We define the *order* of an item in a representation as follows: Objects and constants are order 0. The order of a predicate is one plus the maximum of the order of its arguments. Thus GREATER-THAN(x, y) is first-order if x and y are objects, and CAUSE[GREATER-THAN(x, y), BREAK(x)] is second-order. Examples of higher-order relations include CAUSE and IMPLIES.

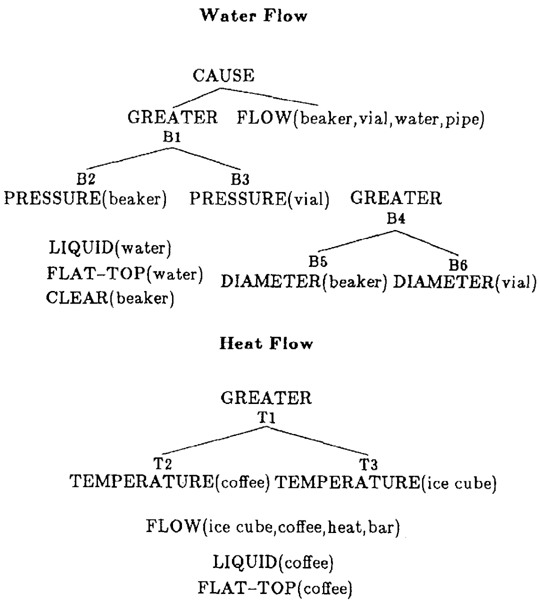


Figure 2.
Simplified Water Flow and Heat Flow Descriptions

might have of these situations (simplified for clarity).

In order to comprehend the analogy "Heat is like water" a learner must:

(1) Set up the object correspondences between the two domains:

```
heat --> water, tube --> metal bar,
beaker --> coffee, vial --> ice cube
```

(2) Discard object attributes, such as `CYLINDRICAL(beaker)`.

(3) Map base relations such as

```
GREATER-THAN[PRESSURE(water, beaker),
             PRESSURE(water, vial)]
```

to the corresponding relations in the target domain.

(4) Observe systematicity: i.e., keep relations belonging to a systematic relational structure in preference to isolated relationships. In this example,

```
CAUSE(GREATER-THAN[PRESSURE(water, beaker),
                   PRESSURE(water, vial)],
      FLOW(water, pipe, beaker, vial))
```

is mapped into

```
CAUSE(GREATER-THAN[TEMPERATURE(heat, coffee),
                   TEMPERATURE(heat,ice cube)],
      FLOW(heat, bar, coffee, ice cube))
```

while isolated relations, such as
`GREATER-THAN[DIAMETER(beaker), DIAMETER(vial)]`
are discarded.

The *systematicity principle* is central to analogy. Analogy conveys a system of connected knowledge, not a mere assortment of independent facts. Prefering systems of predicates that contain higher-order relations with inferential import is a syntactic expression of this tacit preference for coherence and deductive power in analogy. It is the higher-order relational structure that determines which of two possible matches is made. For example, suppose in the previous example we were concerned with objects differing in specific heat, such as a metal ball-bearing and a marble of equal mass, rather than temperatures. Then DIAMETER becomes relevant, since (in a more complete model than we have space for) DIAMETER affects the capacity of a container, the analog to specific heat.

The Structure-Mapping theory has received a great deal of convergent theoretical support in artificial intelligence and psychology. Although there are differences in emphasis, there is widespread agreement on the basic elements of one-to-one mappings of objects with carryover of predicates (Burstein, 1983; Carbonell, 1983; Hofstadter,1984; Kedar-Cabelli, 1985; Reed, 1985; Rumelhart & Norman, 1981; Winston, 1982). Moreover, all these researchers have adopted something like the systematicity principle, or a special case of systematicity. For example, Carbonell focuses on plans and goals as the high-order relations that give constraint to a system, and Winston focuses on causality. Also, some models combine a structure-mapping component, which generates possible interpretations of a given analogy, with a pragmatic component which chooses the relevant interpretation (e.g., Burstein, 1983; Kedar-Cabelli, 1985).

Empirical psychological studies have borne out the prediction that systematicity is a key element of people's implicit rules for analogical mapping. Adults focus on shared systematic relational structure in interpreting analogy. They tend to include relations and omit attributes in their interpretations of analogy, and they judge analogies as more sound and more apt if base and target share systematic relational structure (Gentner, 1980; Gentner & Landers, 1985; Gentner & Stuart, 1983). Finally, in developmental work we have found that children are better at performing difficult mappings when the base structure is systematic (Gentner & Toupin, in press).

Given the existing theoretical and empirical psychological support, we have decided that cognitive simulation is needed to allow us to explore the theory still more deeply.

3. THE STRUCTURE-MAPPING ENGINE: DESIGN

Given the descriptions of a base and a target, SME constructs all syntactically consistent analogical mappings between them. As noted above, the mappings consist of pairwise matches between predicates and objects in the base and target, plus a list of predicates which exist in the base but not the target. This list of predicates is the set of *candidate inferences* sanctioned by the analogy. SME also provides a syntactic evaluation of each mapping. In accordance with Structure-Mapping theory, no domain information beyond the representation of the target is used in SME to evaluate the candidate inferences – that is the job of other modules.

The base and target representations provided to SME are collections of facts called *description groups.* Domain objects and constants are collectively referred to as *entities.* The construction of the analogy is guided by *match rules* which specify which facts and entities in the base and target might match and estimate the believability of each possible component of a match. Importantly, to build a new match function one simply loads a new set of match rules. These rules are the key to SME's flexibility.

An analogy is processed in three steps. First, all potential pairings between items in the base and target are constructed and individually evaluated. Second, all sets of consistent combinations of these pairings are constructed to form the possible global matches and their corresponding candidate inference sets. Finally, the global matches are evaluated syntactically to provide a score. We now describe these computations in detail.

3.1. Step 1: Construct local match hypotheses

SME begins by finding for each entity and predicate in the base the set of entities or predicates in the target that could plausibly match that item. Plausibility is determined by *match hypothesis constructor* rules, which take the form

(MHCrule <*condition*> <*body*>)

The body of these rules is run on each pair of items (one from the base and one from the target) that satisfy the condition and installs a *match hypothesis* which represents the possibility of them matching. For example, we state that all predicates whose predicate name is identical could potentially match with the rule

```
(MHCrule (equal-functors? *base-fact* *target-fact*)
     (install-MH *base-fact* *target-fact*))
```

The likelyhood of each match hypothesis is found by running *match evidence rules* and combining their results. The evidence rules provide support for a match hypothesis by examining the syntactic properties of the items matched. For example, the rule

```
(MHErule (and (equal (mh-type *MH*)   fact)
              (equal-functors? (mh-base-item *MH*)
                               (mh-target-item *MH*)))
     (MHevidence *MH* 0.5 0.0))
```

states "If the two items are facts and their functors are the same, then supply 0.5 evidence in favor of the match hypothesis."* The rules may also examine match hypotheses associated with the arguments of these items to provide support based on systematicity. This causes evidence for a match hypothesis to increase with the amount of higher-order structure supporting it. We use the Dempster-Shafer formalism for probabilities (Shafer, 1976) and combine evidence with a simplified form of Dempster's rule of combination (Prade, 1983; Ginsberg, 1984). By using the simplified formula we are assuming independence among the match hypotheses, but this is not a problem because we are only using it to produce scores for ordering candidates rather than estimating probabilities.

The state of the match between the water flow and heat flow descriptions of Figure 2 after running these first two sets of rules is shown in Figure 3. The weights shown in the figure are the support for each match hypothesis. Internally the program stores a Shafer interval, consisting of the support for the match and the maximum plausible support (i.e., one minus the support against it). The water flow – heat flow analogy is made possible by the program being able to match predicates with different names, such as matching

Match Hypothesis		
Base Node	Target Node	Evidence
$\mathrm{GREATER}_{\mathrm{Pressure}}$	$\mathrm{GREATER}_{\mathrm{Temperature}}$	0.650
$\mathrm{GREATER}_{\mathrm{Diameter}}$	$\mathrm{GREATER}_{\mathrm{Temperature}}$	0.650
$\mathrm{PRESSURE}_{\mathrm{beaker}}$	$\mathrm{TEMPERATURE}_{\mathrm{coffee}}$	0.712
$\mathrm{PRESSURE}_{\mathrm{vial}}$	$\mathrm{TEMPERATURE}_{\mathrm{ice\ cube}}$	0.712
$\mathrm{DIAMETER}_{\mathrm{beaker}}$	$\mathrm{TEMPERATURE}_{\mathrm{coffee}}$	0.712
$\mathrm{DIAMETER}_{\mathrm{vial}}$	$\mathrm{TEMPERATURE}_{\mathrm{ice\ cube}}$	0.712
$\mathrm{FLOW}_{\mathrm{water}}$	$\mathrm{FLOW}_{\mathrm{heat}}$	0.790
$\mathrm{FLAT}_{\mathrm{water}}$	$\mathrm{FLAT}_{\mathrm{coffee}}$	0.790
$\mathrm{LIQUID}_{\mathrm{water}}$	$\mathrm{LIQUID}_{\mathrm{coffee}}$	0.790
vial	ice cube	0.932
beaker	coffee	0.932
water	coffee	0.864
water	heat	0.632
pipe	bar	0.632

Figure 3.
Water Flow – Heat Flow Match After Running Local Rules

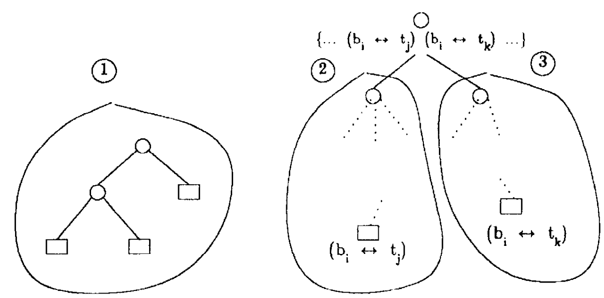
PRESSURE and TEMPERATURE. This behavior is caused by the particular set of rules we are using. In these rules, relational predicates such as GREATER are limited to matching predicates having the same name, while functional predicates such as TEMPERATURE can match other functional predicates. Note that at this stage, SME is entertaining a number of matches that will later be discarded, such as `LIQUID(water)` ↔ `LIQUID(coffee)` and `DIAMETER(vial)` ↔ `TEMPERATURE(ice cube)`.

* Evidence is attributed to a match hypothesis in the form of two numbers. The first number corresponds to evidence in favor of the match and the second number indicates evidence against the match. The sum of these numbers must be less than or equal to one.

3.2. Step 2: Global Match Construction

Once the individual match hypotheses have been constructed and analyzed, SME builds a set of analogical mappings between the base and target. Each mapping is a maximal set of consistent match hypotheses plus the candidate inferences supported by those hypotheses. Consistency is enforced by insisting that a match hypothesis MH is in the analogy only if the mapping includes other match hypotheses that pair up all the arguments of the base and target items of MH. The mappings are maximal in that adding another match hypothesis would lead to a contradiction, as indicated by a base item being matched to two target items or vice versa.

The key to forming the mappings is constructing the sets of entity correspondences (called *Emaps*). Mappings are constructed in four steps. First, find all *entity justifiers*. An entity justifier is a match hypothesis that directly justifies one or more Emaps, in that some of its arguments are entities. Second, associate with each match hypothesis the set of Emaps that it implies. This step is accomplished by propagating Emaps upwards from entity justifiers. The set of Emaps that a match hypothesis supports is simply the union of all Emaps supported by its descendents. Third, create a collection of globally consistent matches, called *Gmaps*. Call a match hypothesis that is not the descendent of any other match hypothesis a *root*. Notice that if the Emaps supported by a root are consistent, then the entire structure under it is consistent. In the simplest case, the entire collection of descendents may be collected together to form a globally consistent match. However, if the root is not consistent, then the same procedure is applied recursively to each descendent. The result is a collection of sets of match hypotheses, within which all Emaps are consistent. The final step is to generate all consistent combinations of these sets, keeping those combinations that are maximal. This is done by first combining Gmaps which are part of



(a)

Gmap #1: { ($\mathrm{GREATER}_{B1}$ ↔ $\mathrm{GREATER}_{T1}$) (FLOW ↔ FLOW)
($\mathrm{PRESSURE}_{B2}$ ↔ $\mathrm{TEMPERATURE}_{T2}$)
($\mathrm{PRESSURE}_{B3}$ ↔ $\mathrm{TEMPERATURE}_{T3}$) }
Emaps: { (beaker ↔ coffee) (vial ↔ ice cube)
(water ↔ heat) (pipe ↔ bar) }
Weight: 0.9800
Candidate Inferences: { (CAUSE $\mathrm{GREATER}_{T1}$ FLOW) }

Gmap #2: { ($\mathrm{GREATER}_{B4}$ ↔ $\mathrm{GREATER}_{T1}$)
($\mathrm{DIAMETER}_{B5}$ ↔ $\mathrm{TEMPERATURE}_{T2}$)
($\mathrm{DIAMETER}_{B5}$ ↔ $\mathrm{TEMPERATURE}_{T2}$) }
Emaps: { (beaker ↔ coffee) (vial ↔ ice cube) }
Weight: 0.0195
Candidate Inferences: { }

Gmap #3: { (LIQUID ↔ LIQUID) (FLAT-TOP ↔ FLAT-TOP) }
Emaps: { (water ↔ coffee) }
Weight: 0.0004
Candidate Inferences: { }

(b)

Figure 4. Gmap Construction

the same base structure (e.g. the Gmap for the pressure inequality would combine with the Gmap for the flow relation to form a single Gmap) and then making any further combinations which are consistent. Figure 4(a) shows how the initial set of Gmaps is formed, while Figure 4(b) shows the final Gmaps created for the water flow – heat flow analogy.

Associated with each Gmap is a (possibly empty) set of candidate inferences. Candidate inferences are base predicates that would fill in structure which is not in the Gmap (and hence not already in the target). In Figure 4(b), for example, Gmap #1 has the top level CAUSE predicate as its sole candidate inference. If the FLOW predicate was not present in the target, then the candidate inferences for a Gmap corresponding to the pressure inequality would be both CAUSE and FLOW. All candidate inferences must be consistent with known target facts. In addition, they must be consistent with the Gmap's structure and supported by some member of it. For example, GREATER-THAN[DIAMETER(coffee), DIAMETER(ice cube)] is not a valid candidate inference for the first Gmap because it does not intersect the existing Gmap structure.

3.3. Step 3: Global Match Selection

Several factors must be taken into account when deciding which Gmap is the best analogy. We have identified three factors as particularly important:

(1) The evidence for the individual match hypotheses in the Gmap.

(2) The candidate inferences sanctioned by the Gmap.

(3) The graph-theoretic structure of the Gmap, e.g., the number and relative size of connected components.

Exploring the relative importance of these and other factors is part of the desiderata for SME, hence we have made the criteria programmable. *Gmap evidence rules*, whose form is much the same as the other kinds of rules mentioned previously, can provide evidence for a Gmap based on whatever factors are deemed appropriate. To make an appropriate selection, evidence for Gmaps is combined under strict adherence to Dempster's rule for combining probabilities. Thus the set of Gmaps is treated as a set of mutually exclusive choices, and evidence in favor of one Gmap implicitly counts as evidence against the others. Dempster's rule automatically normalizes the weights so that the sum of the weights supporting each Gmap will always be less than or equal to one. In Figure 4(b), the Gmap which maps the PRESSURE relation is believed more than the Gmap which maps the DIAMETER relation. This conclusion is based on two rules. The first rule simply permits the evidence for a match hypothesis in a Gmap to count as evidence for that Gmap. The second rule gives evidence of 0.3 to a Gmap for each candidate inference it sanctions.

We suspect tnat the ability to "tune" the criteria for choosing a Gmap will be important for modeling individual differences in analogical style and a subject's domain knowledge. For example, a conservative strategy might favor taking Gmaps with some candidate inferences but not too many, in order to maximize the probability of being right.

4. EXAMPLES

The Structure-Mapping engine has been tested on a number of examples drawn from a variety of domains. We discuss a few examples to further demonstrate SME's flexibility and generality. Our first example is taken from Rutherford's analogy between the solar system and the hydrogen atom. The second example demonstrates how the program reasons about complicated descriptions of water flow and heat flow which were generated by a qualitative reasoning program before the inception of SME.

4.1. Solar System – Rutherford Atom Analogy

The Rutherford model of the hydrogen atom was based on the well-understood behavior of the solar system. Given the descriptions shown in Figure 5, the Structure-Mapping engine constructed three possible interpretations. The most preferred

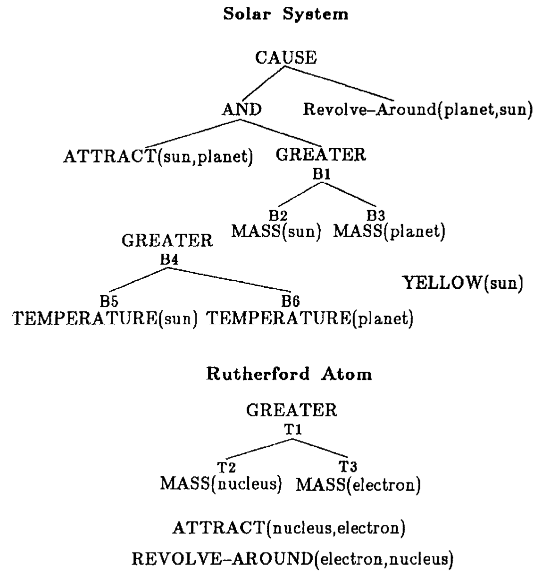


Figure 5. Solar System – Rutherford Atom Analogy

mapping (given a weight of 0.99) pairs up the nucleus with the sun and the planet with the electron. This mapping is based on the mass inequality in the solar system playing the same role as the mass inequality in the atom. It sanctions the inference that the inequality, together with the mutual attraction of the nucleus and the electron, causes the electron to revolve around the nucleus. The other major Gmap (given a weight of 0.01) has the same entity correspondences, but is based on the solar system's temperature inequality mapping to the atom's mass inequality. There is much less belief in this interpretation because the temperature and mass predicates are different and because this Gmap does not allow any candidate inferences. The third Gmap is a spurious collection of match hypotheses which imply that the mass of the sun and planet should correspond to the mass of the electron and nucleus, respectively. There is no higher-level structure to support this interpretation and so the final belief is 1×10^{-6}. This example demonstrates how SME is able to generate all syntactically plausible interpretations of a potentially analogous situation. It also show that our rules have a preference for matching predicates of the same name (e.g. MASS with MASS), but is able to match predicates with different names (e.g. TEMPERATURE with MASS).

4.2. Water Flow – Heat Flow Analogy

The Structure-Mapping engine has applications beyond cognitive simulation. For example, we could use this program in conjunction with a qualitative reasoning program to model the way people use analogy to reason about the physical world. Figure 6 (a) shows a domain description for water flow which was used in an actual qualitative reasoning program (Forbus 1984; Forbus & Gentner, 1983). Figure 6 (b) shows a greatly reduced version of the same program's description of heat flow.

As with the earlier, simplified descriptions of water flow and heat flow, SME was able to make the correct analogical correspondences, creating all of the possible candidate inferences in

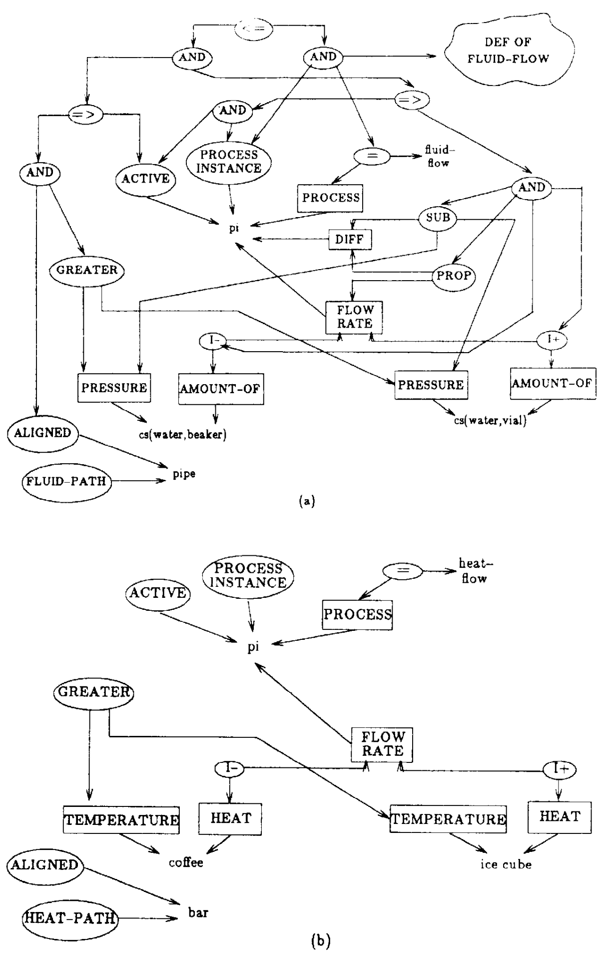


Figure 6. Water Flow (a) and Heat Flow (b)

the process. Interestingly, only one consistent interpretation arose. All other match hypotheses were eliminated because they had no descendants to support their existence. The candidate inferences made were the correct ones, namely that a difference in temperature and an aligned heat path implies an instance of heat flow and that the rate of heat flow between two objects is proportional to the difference in their temperatures.

4.3. Summary

Space limitations forbid a detailed account of our experiments to date; we summarize two here. First, we have analyzed short stories described in predicate calculus to compare mere appearance, surface matches with true analogy. Second, we have begun exploring a number of match algorithms. For example, one set of rules focuses on object attributes (mere-appearance matches), thus mimicking how children tend to treat potentially analogous situations (see below). These rules, when run on the water flow – heat flow descriptions of Figure 2, choose the water to coffee correspondence as the best interpretation due to their surface similarity and fail to notice the relational structure which implies that the role of water actually corresponds to the role of heat in the water flow and heat flow situations.

5. CONCLUSIONS

SME has significant advantages over more traditional matching algorithms. Methodologically, the advantage of producing all possible mappings is that one can easily see syntactically consistent alternatives to the best match. Yet SME's matching algorithm is very efficient, avoiding the extensive backtracking normally associated with pattern-matching systems.* On our large water flow – heat flow example, the program took only 0.7 seconds to perform the entire match on a Symbolics 3640. This includes everything from the construction of local match hypotheses to the gathering of candidate inferences and Gmap construction. The smaller examples average 0.4 seconds. The current program needs to be expanded to properly handle predicates which are commutative (e.g. SUM) or take a variable number of arguments (e.g. AND). In addition, we would like to add the ability to introduce new entities when required by the analogical mapping through the use of Skolem functions.

The results of SME's operation on the examples above provides suggestive evidence concerning a currently debated issue in analogy. The question concerns how much a purely syntactic account of analogy can do. Although many researchers have adopted variants of the systematicity principle, often specific domain knowledge or pragmatic information is used as well. For example, Carbonell (1981, 1983) focuses on plans and goals as the relevant higher-order relations for analogical mapping. Winston's (1982) system uses causal relations in its *importance-guided* matching algorithm. Winston [personal communication, November 1985] has also investigated goal-driven importance algorithms. The extreme view is taken by Holyoak (1985), whose account of analogical mapping relies solely on the relevance of predicates to the current plan. Among the claims of these researchers are (1) purely syntactic information is insufficient to guide analogical mapping and (2) even if it were sufficient, such a system would be inefficient (e.g. Burnstein, 1986, p.358). The evidence from SME so far suggests otherwise, since it generates intuitively plausible answers and does so rapidly. We intend to explore this issue more fully by using a variety of examples to see if and when the purely syntactic approach breaks down. Clearly content knowledge must be invoked at some point to evaluate whether the candidate inferences from a given analogy are appropriate. This suggests a model which uses a context-sensitive, expectation-driven system to evaluate the output of SME. This extension is compatible with the combination models proposed by Burstein (1983) and Kedar-Cabelli (1985).

In addition to tests of the basic algorithm, we plan several cognitive simulation studies of analogical reasoning and learning. We mention only one here. Psychological research shows a marked developmental shift in analogical processing. Young children rely on surface information in analogical mapping; at older ages, systematic mappings are preferred (Gentner & Stuart, 1983; Gentner & Toupin, in press; Holyoak, Juin & Billman, 1985; Vosniadou, 1985). Further, there is some evidence that a similar shift from surface to systematic mappings occurs in the novice-expert transition in adults (Chi, Glaser & Reese 1982; Larkin, 1985; Novick, 1985; Reed, 1985; and Ross, 1984).

In both cases there are two very different interpretations for this analogical shift: (1) acquisition of knowledge; or (2) a change in the analogy algorithm. The knowledge-based interpretation is that children and novices lack the necessary higher-order relational structures to guide their analogizing. The second explanation is that the algorithm for analogical mapping changes, either due to maturation or learning. In human learning it is difficult to decide this issue, since exposure to domain knowledge and practice in analogy and reasoning tend to occur simultaneously. SME gives us a unique opportunity to vary independently the analogy algorithm and the amount and kind of domain knowledge. For example, we can compare identical evaluation algorithms operating on novice versus

* While we have not yet explored this possibility, it appears that a variant of this matching algorithm could be very useful for connectionist architectures.

expert representations, or we can compare different analogy evaluation rules operating on the same representation (see summary above). The performance of SME under these conditions can be compared with novice versus expert human performance.

ACKNOWLEDGEMENTS

The authors wish to thank Janice Skorstad for invaluable assistance in encoding domain models.

REFERENCES

Anderson, J., *The Architecture of Cognition*, Harvard University Press, Cambridge, Mass, 1983.

Buckley, S., *Sun up to sun down*, McGraw-Hill Company, New York, 1979

Burstein, M. H., Concept formation by incremental analogical reasoning and debugging. *Machine Learning: An Artificial Intelligence Approach Vol. II*, R.S.Michalski, J.G.Carbonell, and T.M.Mitchell (editors), Morgan Kaufmann, 1986.

Carbonell, J. G. Learning by analogy: Formulating and generalizing plans from past experience. *Machine Learning*, Michalski, R. S., Carbonell, J., and Mitchell, T. (Eds.), Tioga Publishing Company, Palo Alto, California, 1983.

Carbonell, J. G., Derivational analogy in problem solving and knowledge acquisition. *Proceedings of the Second International Machine Learning Workshop*, University of Illnois, Monticello, Illinois, June, 1983.

Chi, M. T. H., Glaser, R., & Reese, E. Expertise in problem solving. In R. Sternberg (Ed.), *Advances in the psychology of human intelligence* (Vol.1). Hillsdale, N.J., Erlbaum, 1982.

Forbus,K.D., Qualitative Process Theory. MIT Artificial Intelligence Laboratory Technical Report No. 789, July, 1984.

Forbus,K.D., and D.Gentner, Learning Physical Domains: Towards a Theoretical Framework. *Machine Learning: An Artificial Intelligence Approach Vol. II*, R.S.Michalski, J.G.Carbonell, and T.M.Mitchell (editors), Morgan Kaufmann, 1986.

Gentner, D., *The structure of analogical models in science*. BBN Tech. Report No. 4451, Cambridge, MA., Bolt Beranek and Newman Inc., 1980

Gentner, D., Are scientific analogies metaphors?. In Miall, D., *Metaphor: Problems and perspectives*, Harvester Press, Ltd., Brighton, England, 1982

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

Gentner D., & Gentner, D. R., Flowing waters or teeming crowds: Mental models of electricity, In D. Gentner & A. L. Stevens, (Eds.), *Mental Models*, Erlbaum Associates, Hillsdale, N.J., 1983

Gentner, D., & Landers, R. Analogical reminding: A good match is hard to find. In *Proceedings of the International Conference on Systems, Man and Cybernetics*. Tucson, Arizona, 1985.

Gentner, D., & Stuart, P. Metaphor as structure-mapping: What develops. Bolt Beranek and Newman Technical Report No. 5479, Cambridge, Massachusetts, 1983.

Gentner, D. & Toupin, C. (in press). Systematicity and surface similarity in the development of analogy. *Cognitive Science*.

Ginsberg,M.L., Non-Monotonic Reasoning Using Dempster's Rule. *Proceedings AAAI*, August, 1984.

Hofstadter, D. R. The Copycat project: An experiment in nondeterministic and creative analogies. M.I.T. A.I. Laboratory memo 755. Cambridge, Mass: M.I.T., 1984.

Holyoak, K. J. The pragmatics of analogical transfer. In G. H.Bower (Ed.) *The psychology of learning and motivation. Vol.1.* New York: Academic Press, 1984.

Holyoak, K. J., Juin, E. N. & Billman, D. O. (in press). Development of analogical problem-solving skill. *Child Development.*

Kedar-Cabelli, S. (1985). Purpose-directed analogy. *Proceedings of the Seventh Annual Conference of the Cognitive Science Society*, Irvine, CA.

Larkin, J. H. Problem representations in physics. In D. Gentner & A. L. Stevens (Eds.) *Mental Models.* Hillsdale, N. J., Lawrence Erlbaum Associates, 1983.

Novick, L. R. Transfer and expertise in problem solving: A conceptual analysis. Stanford University: Unpublished manuscript, 1985.

Prade,H., A Synthetic View of Approximate Reasoning Techniques. *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, 1983.

Reed, S. K. A structure-mapping model for word problems. Paper presented at the meeting of the Psychonomic Society, Boston, 1985.

Ross, B. H. Remindings and their effects in learning a cognitive skill. *Cognitive Psychology*, **16**, 371-416, 1984.

Rumelhart, D. E., & Norman, D. A. Analogical processes in learning. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition*, Hillsdale, N.J. Erlbaum, 1981.

Shafer,G., A Mathematical Theory of Evidence, Princeton University Press, Princeton, New Jersey, 1976.

Van Lehn, K. & Brown, J. S. Planning nets: A representation for formalizing analogies and semantic models of procedural skills. In R. E. Snow, P. A. Federico & W. E. Montague (Eds.), *Aptitude, learning and instruction: Cognitive process analyses.* Hillsdale, N. J. Erlbaum, 1980.

Van Lehn, K., Felicity Conditions for Human Skill Acquisition: Validating an AI-based Theory. Xerox Palo Alto Research Center Technical Report CIS-21, 1983.

Vosniadou, S. On the development of metaphoric competence. University of Illinois: Manuscript submitted for publication, 1985.

Winston, P. H. Learning and reasoning by analogy. *Communications of the ACM*, 23(12), 1980.

Winston, P. H. Learning new principles from precedents and exercises. *Artificial Intelligence*, **19**, 321-350, 1982.