Yan, J., Forbus, K., and Gentner, D. (2003). A theory of rerepresentation in analogical
matching. Proceedings of the Twenty-fifth Annual Meeting of the Cognitive Science Society.

A theory of rerepresentation in analogical matching

Jin Yan (j-yan@northwestern.edu)
Kenneth D. Forbus (forbus@northwestern.edu)
Qualitative Reasoning Group, Northwestern University, 1890 Maple Avenue
Evanston, IL, 60201, USA
Dedre Gentner (gentner@northwestern.edu)
Psychology Department, Northwestern University, 2029 Sheridan Road
Evanston, IL, 60208-2710, USA

Abstract

Psychologically, rerepresentation appears to be an important technique for achieving flexibility in analogical matching. This paper presents a concise theory of rerepresentation in analogical matching. It divides the problem into *detecting opportunities* for rerepresentation, *generating rerepresentation suggestions* based on libraries of general methods, and *strategies* for controlling the rerepresentation process. We show that the kinds of opportunities can be exhaustively derived from the principles of structure-mapping, and the methods for detecting them derived from consideration of how the SME algorithm works. Four families of rerepresentation methods are proposed, as well as task-independent and task-dependent constraints on strategies. Implemented simulation examples are used for illustration.

Introduction

Rerepresentation re-construes parts of compared situations in order to improve a match. It is an important process in analogical reasoning and learning. In development, rerepresentation appears to play an important role in learning. For example, Kotovsky & Gentner (1996) found that children are better able to make cross-dimensional analogies when they have been induced to rerepresent the two situations to permit noticing the common magnitude increase. Rerepresentation also plays an important role in scientific discovery. For example, Gentner *et al* (1997) argue that representation played a crucial role in Kepler's working through his analogy of *vis-Motrix* to light.

This paper presents a concise theory of rerepresentation in analogical matching. The next section outlines the computational issues surrounding rerepresentation. Our theory of rerepresentation is described next, and illustrated with implemented examples from a computer simulation using the Structure-Mapping Engine (SME) [Falkenhainer *et al* 1986, 1989; Forbus *et al* 1994][1]. Finally we discuss related and future work.

[1] The representational vocabulary is drawn from Cycorp's Cyc knowledge base, plus our own extensions. Opportunity detection is carried out by Lisp code that uses SME datastructures, and suggestions about applicable methods are generated using our FIRE reasoning engine.

Rerepresentation in analogical reasoning

We assume, as usual in analogical reasoning research, that the representations used in matching are internal descriptions, as opposed to, for instance, lexical items.

Every analogical matcher must, as part of its operation, make decisions about whether or not two local items (statements or entities) within the descriptions it is comparing can be aligned. Structure-mapping postulates that these decisions are made based on *tiered identicality*, i.e., that relationships must by default be identical, and only under special circumstances should looser criteria (such as *minimal ascension* [Falkenhainer 1990]) be used to sanction local matches. Other models have postulated that more generous criteria are always used, such as ignoring the semantics provided by the relations, yielding a purely structural match (e.g., IAM [Keane 1990]) or using some other representational resource to determine whether two relationships are alignable (e.g., the use of WordNet in [Holyoak & Thagard, 1989]). Computationally, the tradeoff is between false negatives and false positives: Stricter criteria will miss potential matches, but looser criteria will generate more false positives.

To evaluate the plausibility of where human analogical processing lies on this tradeoff, it is useful to consider how analogical matching fits into the larger scheme of cognitive processing. Functionally, there are processes that generate the descriptions used as the base and target descriptions to be matched. This includes the encoding processes used to construct representations from perceptual information, memory processes used to retrieve specific experiences and general knowledge from long-term memory, and reasoning that we might be doing upon such information, e.g., during problem solving. While such processes are variable, they in fact involve a large degree of regularity: Whatever internal representation is generated for seeing, for example, a cat is expressed in the same internal representational conventions from one instant to the next, although the details of the specific descriptions computed may change as the cat stretches. Similarly, the descriptions retrieved by memory use a uniform set of representational conventions. The specific contents of descriptions for two distinct cats, for example, might vary widely due to differences in what was attended to as well as differences between the cats themselves, but it seems likely that much of the basic vocabulary of perceptual and physical relationships is roughly constant over time. On the other hand, differences in attention and task demands will affect what is encoded and to some degree how, and learning can change conceptual vocabularies and encoding strategies (cf. [Chi *et al* 1981]). Information gleaned from language can be highly variable (e.g., verb choices such as “ambled” versus “strolled” versus “ran” presumably affect internal representations beyond the difference in lexemes), and how much canonicalization occurs when understanding language is still an open question.

This analysis suggests that false negatives are less of a concern than false positives, especially for more concrete descriptions. Furthermore, false positives put more burden on matchers: More correspondences must be produced, and more possibilities considered when merging local hypotheses. Given that merge operations are approximations (otherwise they would involve implausible amounts of backtracking search) and there are likely to be resource limitations on the amount of correspondences that can be generated, avoiding false positives seems like a better strategy for the organism.

Rerepresentation seems inevitable, given that encoding processes can be variable, inputs vary, and representations evolve over time. The real question is, where should it occur? The structure-mapping model is to place it outside the matcher itself. Consider the process(es) that evaluate the matcher’s output. The mapping(s) produced must be examined to see if they yield results that are useful for the current task. If they do not, changes ranging from tweaking the content of the base and target (i.e., rerepresentation) to choosing a new base or even abandoning the current line of effort are options available to such processes. This seems to us to be a natural place to recover from false negatives. The mapping(s) can be examined for *opportunities* for rerepresentation. A library of *rerepresentation methods*, based on the type of opportunity, provide suggestions for specific rerepresentations. Task-specific *rerepresentation strategies* determine which suggestions, if any, should be acted upon. Once rerepresentation(s) have been made, changing the base and target, the match can be updated and the results evaluated again.

A structure-mapping theory of rerepresentation

We divide our account of rerepresentation into three parts: (1) *Detecting opportunities* for rerepresentation, (2) *methods* for rerepresentation, and (3) *strategies* that control which opportunities are exploited and what methods are used. We discuss each in turn.

Opportunities

We characterize opportunities for rerepresentation based on which constraints of structure-mapping are violated. Recall that, in addition to tiered identicality, the constraints of *structural consistency* define what legal matches are:

- *1:1 constraint:* Each item in the base maps to at most one item in the target, and vice-versa.
- *Parallel connectivity constraint:* If a correspondence between two statements is included in a mapping, then so must correspondences between its arguments.

Violations of identicality and 1:1 are more fundamental; as shown below, whether or not parallel connectivity is violated depends on where the failure to satisfy these other constraints occurs. Table 1 concisely describes the possibilities. We discuss each in turn, describing how to detect them based on the representations used in SME.

Table 1: Rerepresentation opportunities

Constraint	Violates parallel connectivity?	Opportunity
Identicality	Yes	Holes
	No	Gulches
1:1	Yes	Rivals
	No	Leftovers

Holes Recall that the initial step of the SME algorithm involves finding, in parallel, local *match hypotheses* that represent potential correspondences between items in the base and target descriptions. What statements are initially aligned is governed by the *tiered identicality* constraint; by default only identical relations are matched. Hence initially match hypotheses are constructed between all pairs of statements from base and target that have identical relations. The *parallel connectivity* constraint requires that corresponding arguments be aligned for the correspondence between two statements to be structurally consistent. Consequently, match hypotheses are also installed between arguments of aligned statements, if doing so would not violate tiered identicality. (Non-identical functions can be aligned, as can any pair of entities.) When this process is complete, the match hypothesis forest so produced serves as the starting point for grouping maximal structurally consistent clusters of match hypotheses into *kernel mappings*, which are combined via a greedy merge algorithm to produce mappings.

Holes arise due to failures of the process of aligning arguments. Consider the following pair of statements:

B1:
```
(cause (walk John Cave)
       (inside John Cave))
```
T1:
```
(cause (run Jill Chamber)
       (inside Jill Chamber))
```

SME would construct a match hypothesis between B1 and T1, based on the identical relationships. This in turn would cause it to attempt to construct match hypotheses between corresponding arguments. It would succeed for the consequents, since the relations are identical. It would fail for the antecedent, since `walk` and `run` are different relationships, assuming strict identicality. Thus the hypothesis that B1 and T1 can match is marked as structurally inconsistent. This failure is an example of a hole. Holes can thus be detected by finding structurally inconsistent match hypotheses whose failure is due to an argument misalignment. SME records such information when marking a match hypothesis as structurally inconsistent, making detection easy.

Consider for example part of a description of two physical situations involving flow, water flow and heat flow (adapted from [Buckley, 1979]):

```
B2: (cause (higherPressure Beaker Vial)
           (flow Beaker Vial Water Pipe))
T2: (cause (hotterThan Coffee IceCube)
           (flow Coffee IceCube Heat Bar))
```

The higherPressure and hotterThan arguments are not alignable because they are not identical. Such domain-specific relationships appear to be used early in development [Kotovsky & Gentner, 1996]. Similarly, from [Clement & Gentner, 1991],

```
B3: (implies (slurps Tam Minerals)
             (attachesTo Tam Rock))
T3: (implies (records Satellite Sounds)
             (orbits Satellite Planet))
```

Here both the antecedent and consequent fail to align, because they are very domain-specific. Below we will see how such mismatches can be overcome.

Gulches: The only way that identicality can cause a failure to match two items without causing a hole is if the items are not themselves the arguments of any other pair of matching items, i.e., one or both are top-level expressions (aka *roots*) of their respective descriptions. Such statements in the base show up as *candidate inferences* of the match. Gulches can be detected by looking for roots in the base and the target whose arguments have structurally consistent match hypotheses but do not themselves match.

Consider for example a fragment from a variation of the classic solar system/Rutherford atom analogy:

```
B4: (causes
     (and (greaterThan (Mass Sun)
                       (Mass Planet))
          (attracts Sun Planet))
     (revolveAround Planet Sun))
T4: (implies
     (and (greaterThan (Mass Nucleus)
                       (Mass Electron))
          (attracts Nucleus Electron))
     (revolveAround Nucleus Electron))
```

Because the root statements themselves do not match, we have a gulch.

Rivals: Rivals are violations of the 1:1 constraint that lead to structural inconsistency of at least one match hypothesis. This occurs when different correspondences for the same entity are implied by the match hypotheses for a statement's arguments. (SME records such information during its structural consistency calculations.)

For example, consider matching a general schema for a feedback controller against a specific feedback system, a thermostat [Ma, 1999]. Here is a small fragment of the representations involved:

```
B5: (senses SensorX SensedParameter)
    (compares ComparatorX SensedParameter
              SetpointX)
T5: (senses ThermostatY (Temperature AirY))
    (compares ThermostatY (Temperature AirY)
              ThresholdY)
```

In fact, the thermostat plays the role of both the comparator and sensor in the abstract schema. However, this match cannot be allowed, since it violates the 1:1 constraint.

Leftovers: Mappings are constructed by combining kernels, using a greedy merge process [Forbus *et al* 1994]. This process starts with the largest kernel, and adds as many kernels to it as possible, subject to maintaining the 1:1 constraint. (Notice that, since kernels are already structurally consistent and maximal, merging two kernels cannot violate any other structure-mapping constraint.) *Leftovers* are kernels that are left out of a mapping because they have one or more entity correspondences that are inconsistent with the mapping.

Typically leftovers are unfixable, since they represent fundamentally different construals of the same comparison. However, sometimes they indicate that a change in reification can improve a match. Consider for example matching a description of a car and a motorcycle, where the tires of each are explicitly described as distinct individuals. Only two tires of the car can be involved in such a match, since each can only match to one tire of the motorcycle, e.g.

```
B6: (isa LeftFrontWheel Wheel)
    (isa RightFrontWheel Wheel)
    (isa LeftRearWheel Wheel)
    (isa RightRearWheel Wheel)
    (hasAttributes LeftFrontWheel RoundShape)
    (hasAttributes RightFrontWheel RoundShape)
    (hasAttributes LeftRearWheel RoundShape)
    (hasAttributes RightRearWheel RoundShape)
T6: (isa FrontWheel Wheel)
    (isa RearWheel Wheel)
    (hasAttributes FrontWheel RoundShape)
    (hasAttributes RearWheel RoundShape)
```

Given relational structure that ties wheels to their function (i.e., rear wheels to providing power, front wheels for steering, depending on the car) the motorcycle's front and rear wheels will be matched to one or the other of the car's front and rear wheels, randomly. The unmatched wheels are leftovers.

Completeness. Are there other opportunities for rerepresentation beyond those listed here? Our analysis suggests not. The only constraint of structure-mapping theory we have not exploited is systematicity. But systematicity is a preference, providing guidance as to better or worse choices rather than ruling some out, as the others do. Since the opportunities described in Table 1 exhaust the constraints of structure-mapping, we conclude that this set is complete.

Now that we have characterized the opportunities for rerepresentation, we can examine methods for using them.

Methods

The appropriate rerepresentation method for each type of opportunity depends on the principle constraint being violated in it (i.e., identicality versus 1:1). While the set of opportunities is fixed, deriving from the nature of the constraints of structure-mapping, the set of rerepresentation methods is relatively open. Nevertheless, we can characterize families of methods for each type, as shown in Table 2. We describe each in turn.

Table 2: Rerepresentation strategies

Constraint	Methods
Identicality	Transformation Decomposition …
1:1	Entity splitting Entity Collecting …

Transformation:

Transformations are rewrite rules that transform one or both of a pair of statements comprising a hole or gulch into equivalent statements that have the same meaning, at least with respect to the current description. For example,

B7: `(greaterThan (Gravity Sun) (Gravity Earth))`

T7: `(lessThan (Gravity Earth) (Gravity Sun))`

can be brought into alignment by transforming T7 to the predicate `greaterThan` and reversing the arguments. Some transformations are more extensive, i.e.,

B8: `(higherPressure Beaker Vial)`

T8: `(hotterThan Coffee IceCube)`

from the earlier example requires rewriting both expressions in terms of a more general, dimensional-independent comparative (e.g., `greaterThan`) and encoding the dimension by functions, e.g.,

B8':
```
(greaterThan (Pressure Beaker)
             (Pressure Vial))
```

T8':
```
(greaterThan (Temperature Coffee)
             (Temperature IceCube))
```

which will match because non-identical function matches are allowed by structure-mapping, precisely to support these kinds of cross-dimensional comparisons. This strategy was proposed by Kotovsky and Gentner (1996) as part of the explanation for why children improve in their ability to notice cross-dimensional matches after experiencing a series of close comparisons.

Decomposition:

Transformations are truth-preserving, but sometimes the relational structure supported by a statement only requires some aspect of its meaning. In the walk/run case above, it is the underlying commonality that movement is occurring that is important; the mover is now inside the place they were moving to. *Decomposition strategies* use the axioms that provide the meaning of relations to identify common aspects of their meaning, which can then be used in place of the original relationship. Thus in the B1/T1 example above we might have

B1':
```
(cause (moveTo John Cave)
       (inside John Cave))
```

T1':
```
(cause (moveTo Jill Chamber)
       (inside Jill Chamber))
```

Similarly, in the Tams/Satellite example above, if we view `slurps` and `records` as having a common relational component of `collects`, and `attachesTo` and `orbits` as having a common relational component of `connectsTo`, we would have via decomposition:

B3':
```
(implies (collects Tam Minerals)
         (connectsTo Tam Rock)
```

T3':
```
(implies (collects Satellite Sounds)
         (connectsTo Satellite Planet))
```

Entity splitting:

Often the same entity plays multiple roles in the same representation. Consider again the thermostat example. The conflict arises because it is playing two distinct roles (*sensor* and *comparator*) in the functional description. If we refine the description of the thermostat, observing that it is the curvature of the bimetallic strip that measures the temperature, and the angular distance between the bimetallic coil and the dial's angle that provides the comparison, then each of these aspects of the thermostat can match to distinct functional descriptions. This is an example of an *entity splitting* strategy. In general, entity splitting strategies require identifying ways to divide up an entity into distinct parts or aspects, and rewrite its roles in the description to use one or the other of these parts or aspects.

In the example of functional matching of a thermostat raised earlier, examining the parts of the thermostat yields two distinct components responsible for different aspects of the functionality, e.g.,

T5':
```
(senses (CurvatureFn BimetallicStrip)
        (Temperature AirY))
(compares (AngleFn BimetallicStrip)
          (Temperature AirY)
          ThresholdY)
```

Each of these components now matches to a different part of the functional specification.

Entity collecting:

Often there are multiple entities that play equivalent roles in some representation, such as the tires on a car, the strands in a DNA molecule, and the players on a team. Consider such corresponding collections in the base and in the target. If they are equivalent with respect to the current description, there will be match hypotheses connecting each pair, although any mapping will select only a subset of these matches. If the cardinality of the two collections is different, then some will be left out in any mapping. A mapping could thus be improved by reifying these collections as explicit sets, and stating properties formerly associated with distinct individuals as properties of the sets. This brings more relational structure to bear on corresponding entities, and hence will raise the structural evaluation of the match. We call such strategies *entity collecting* strategies. In general, once a cluster of rival entity match hypotheses has been identified, knowledge about the kinds of entities involved must be used ascertain whether or not they can be reified into a collection (e.g., the strands of a DNA molecule or the players on a team), and to identify how statements about the individuals can (or cannot) be applied to the collection. Entity collection does not always make sense: If most of the relational structure in the description concerns differentiating the roles that each team member plays, for example, replacing the player descriptions with a set of players would be unwise.

Consider again the car/motorcycle example described earlier. If we assume a function `WheelsFn` that denotes the set of wheels something has, and relationships that distribute collection membership and attributes over set membership (`membersIsa` and `membersHaveAttribute`)

```
B6': (membersIsa (WheelsFn MyCar) Wheel)
     (membersHaveAttribute (WheelsFn MyCar)
                           RoundShape)
T6': (membersIsa (WheelsFn MyMotorcycle) Wheel)
     (membersHaveAttribute (WheelsFn MyMotorcycle)
                           RoundShape)
```

Strategies

Conceptually, we view the process of rerepresentation as occurring in the following steps:

1. Opportunities for rerepresentation are detected using the criteria described above, and selected for further processing.
2. For each opportunity, methods are retrieved and tried to see if they can provide an improvement. Each such improvement is a *rerepresentation suggestion*.
3. One or more suggestions is adopted, causing changes in the base and/or target.
4. The match is re-performed with the updated base and target descriptions.
5. The process continues until the match is suitable.

Strategies for controlling the rerepresentation process depend heavily on context and task demands. These factors determine three things about the process: (1) when the result of a mapping is satisfactory for current purposes, and rerepresentation (or further rerepresentation) can be ignored, (2) when the process should be aborted, in favor of trying a new base or target, or something else entirely, and (3) which of the possible structures that could be added to a match via rerepresentation would be preferable (e.g., might provide a desired candidate inference).

However, we also assume that the following task-independent factors hold for human rerepresentation strategies: (1) *Systematicity:* all else being equal, rerepresentation suggestions that lead to larger structural evaluation scores will be preferred. This is simply the extension of the systematicity preference of structure-mapping to rerepresentation. (2) *High selectivity:* The selection process is tightly controlled, so that very few of the possible opportunities are selected for consideration. As with the preference criteria for selecting which suggestions are adopted, we believe that this choice is governed by a combination of structural evaluation and task-specific criteria.

The high degree of dependence on context and task makes meaningful simulation of the overall strategic process in isolation difficult. Consequently, we have focused our simulation efforts on opportunity detection and rerepresentation methods, as demonstrated above, and postpone simulation of strategies to future work.

Related Work

The theory of rerepresentation presented here relies mainly on the concepts of structure-mapping theory; therefore to the extent that other accounts and models use the constraints of structure-mapping theory, it could be adapted to them, although the specific methods for detecting opportunities would have to be changed, since those rely on the processing model of SME as well.

Most models of analogical matching (cf. IAM [Keane 1990], LISA [Hummel & Holyoak, 1997]) have never been used as components in larger simulations, relying entirely on hand-generated representations. By contrast, SME has been used as a module in a variety of larger simulations and performance systems, and has demonstrated the ability to work with descriptions created automatically from large-scale knowledge bases created by others (cf. [Mostek *et al* 2000][Forbus, 2001][Forbus, *et al* 2002]). LISA and DRAMA's [Eliasmith & Thagard 2001] inability to match more than a handful of relationships seems problematic, given the ability of people to match everyday visual and linguistic material that is significantly more complex.

The most closely related work on rerepresentation is that of Hofstader's FARG group, with systems such as CopyCat [Hofstader & Mitchell, 1994] and TableTop [French, 1995] which combined matching with inference systems to construct representations. The matchers in both CopyCat and TableTop were domain-specific; in contrast SME is domain-independent.

Finally, representation transformation similar to those described here are sometimes used in case-based reasoning systems that rely on structured representations (cf. [Kolodner, 1994] [Leake, 1996]). In CBR systems these transformations are used to adapt case knowledge to the current situation directly, in contrast with our use of them to improve the match itself.

Discussion

Previous work has shown that rerepresentation is an important aspect of analogical reasoning and learning. This paper presents a general theory of rerepresentation. It divides the problem into detecting opportunities, methods which suggest rerepresentations based on opportunities, and strategies that organize the application of the suggestions. Because we were able to derive the kinds of opportunities directly from the theoretical constraints of structure-mapping, we claim the set we propose here completely characterizes them. On the other hand, the methods for rerepresentation, which depend on what constraint is violated, are somewhat more open, since they depend on the specific content of the representation. However, even here we were able to identify four families of methods that we believe covers a broad range of rerepresentation phenomena. Some of these have been identified in the literature before, but our linking them into a tight theoretical framework is novel. Finally, we discussed strategies for rerepresentation. Since, according to our theory, strategies are strongly dependent on context and task, there are few constraints on them that can be derived directly from a general theory of rerepresentation (unlike opportunities and methods), but were still able to propose two constraints on them (*systematicity* and *high selectivity*). Evidence for the utility of this theory was provided via simulation examples drawn from the literature involving opportunity detection and the construction and application of rerepresentation suggestions.

Our next step is to expand our implementation. Currently opportunity detection is fully implemented, but the library of rerepresentation methods contains only representative samples from each of the categories. We plan to expand this library to handle the full range of rerepresentation problems we have encountered in our simulation work. Without the contextual and task constraints of a larger simulation to constrain strategy, the choice of what rerepresentation suggestions are followed is entirely by hand. Thus we see another important step to be embedding our current rerepresentation implementation into a larger-scale simulation, to see how well we can model phenomena from developmental and conceptual change research. This effort will help us to develop a more detailed account of the strategies of rerepresentation.

Acknowledgments

This research was supported by the Office of Naval Research. We thank Ken Kurtz, Thomas Hinrichs, Sven Kuehne and Bryan Quinn for insightful comments and discussion.

References

Buckley, S. 1979. *From Sun Up to Sun Down:Understanding Solar Energy*. McGraw-Hill.

Chi, M. T. H., Feltovich, P. J., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science*, 5, 121-152.

Clement, C. A., & Gentner, D. (1991). Systematicity as a selection constraint in analogical mapping. Cognitive Science, 15, 89-132.

Eliasmith, C. & P. Thagard. (2001) Integrating Structure and Meaning: A Distributed Model of Analogical Mapping. *Cognitive Science*.

Falkenhainer, B. (1990). Analogical interpretation in context. In *The Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*. Cambridge, MA: Lawrence Erlbaum Associates.

Falkenhainer, B., Forbus, K., and Gentner, D. ``The Structure-Mapping Engine'' Proceedings of AAAI-86, Philadelphia, PA, August, 1986

Falkenhainer, B., Forbus, K., Gentner, D. (1989) The Structure-Mapping Engine: Algorithm and examples. *Artificial Intelligence*, 41, pp 1-63

Forbus, K. 2001. Exploring analogy in the large. In Gentner, D., Holyoak, K. and Kokinov, B. (Eds) *The Analogical Mind: Perspectives from Cognitive Science*. Cambridge, MA: MIT Press.

Forbus, K., Ferguson, R. and Gentner, D. (1994) Incremental structure-mapping. *Proceedings of the Cognitive Science Society,* August.

Forbus, K., Mostek, T. and Ferguson, R. 2002. An analogy ontology for integrating analogical processing and first-principles reasoning. Proceedings of IAAI-02, July 2002.

French, R. M. (1995). *The subtlety of similarity.* Cambridge, MA: The MIT Press.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7, 155-170.

Gentner, D., Brem, S., Ferguson, R. W., Markman, A. B., Levidow, B. B., Wolff, P., & Forbus, K. D. (1997). Analogical reasoning and conceptual change: A case study of Johannes Kepler. *The Journal of the Learning Sciences, 6*(1), 3-40.

Hofstadter, D. R., & Mitchell, M. (1994). The Copycat project: A model of mental fluidity and analogy-making. In K. J. Holyoak & J. A. Barnden (Eds.), *Advances in connectionist and neural computation theory: Vol. 2. Analogical connections* (pp. 31-112). Norwood, NJ: Ablex.

Holyoak, K. J., & Thagard, P. (1989). Analogical mapping by constraint satisfaction. *Cognitive Science*, 13(3), 295-355.

Hummel, J. E., & Holyoak, K. J. (1997). LISA: A computational model of analogical inference and schema induction. *Psychological Review.*

Keane, M. T. (1990). Incremental analogising: Theory & model. In K. J. Gilhooly, M. T. G. Keane, R. H. Logie, & G. Erdos (Eds.), *Lines of thinking*. Chichester, England: Wiley

Kolodner, J. L. (1994). *Case-based reasoning*. San Mateo, CA: Morgan Kaufmann Publishers.

Kotovsky, L., & Gentner, D. (1996). Comparison and categorization in the development of relational similarity. *Child Development, 67*, 2797-2822.

Leake, D. (Ed.) 1996. *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. MIT Press.

Ma, Joyce. 1999. A Case study of Student Reasoning about Feedback Control in a Computer-based Learning Environment. 29th ASEE/IEEE Frontiers in Education Conference, San Juan, Puerto Rico

Mostek, T., Forbus, K. and Meverden, C. 2000. Dynamic case creation and expansion for analogical reasoning. *Proceedings of AAAI-2000*. Austin, Texas.