Integrating Analogical Mapping and General Problem Solving: The Path-Mapping Theory

Dario D. Salvucci
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
dario+@cs.cmu.edu
(412) 268-8102

John R. Anderson
Department of Psychology
Carnegie Mellon University
Pittsburgh, PA 15213
ja+@cmu.edu
(412) 268-2788

July 22, 1999

Abstract

This article describes the path-mapping theory of how humans integrate analogical mapping and general problem solving. The theory posits that humans represent analogs with declarative roles, map analogs by lower-level retrieval of analogous role paths, and coordinate mappings with higher-level organizational knowledge. Implemented in the ACT-R cognitive architecture, the path-mapping theory allows models of analogical mapping behavior to incorporate and interface with other problem-solving knowledge. Path-mapping models thus can include task-specific skills such as encoding analogs and generating responses, and can make behavioral predictions at the level of real-world metrics such as latency or correctness. We show that the path-mapping theory can successfully account for the major phenomena addressed by previous theories of analogy. We also demonstrate a path-mapping model that can account for subjects' incremental eye-movement and typing behavior in a story-mapping task. We discuss extensions and implications of this work to other areas of analogy and problem-solving research.

Introduction

Analogy — the process of finding and using correspondences between concepts — plays a fundamental and ubiquitous role in human cognition. From mathematical problem solving (Novick & Holyoak, 1991) to computer programming (Anderson & Thompson, 1989) to creative discovery (Holyoak & Thagard, 1995), analogy facilitates better understanding of old knowledge and the formation and inference of new knowledge. Given the importance of analogy to cognition, it has become clear that any general theory of cognition must account for or incorporate analogical reasoning. Some researchers are working toward a general cognitive theory starting from a theory of analogy (e.g., Gentner, 1989; Hummel & Holyoak, 1997; Keane, Ledgeway, & Duff, 1994), while others are working to understand analogy within a general theory of cognition (Anderson & Thompson, 1989; Salvucci & Anderson, 1998a).

The centerpiece of analogy, and the primary focus of this paper, is the process of analogical mapping. For a given pair of concepts, analogical mapping determines the best correspondence between objects and relations in one domain and objects and relations in another domain. For instance, in Duncker's (1945) tumor-ray problem, a doctor removing a tumor can be mapped to a general attacking a fortress. While other subprocesses play a role in the analogy process (e.g., access: Ross, 1987, 1989; induction: Gick & Holyoak, 1983; transfer: Catrambone, 1994; adaptation: Novick & Holyoak, 1991), analogical mapping has stood out as a fundamental yet complex component of analogy that warrants detailed study.

This paper presents the path-mapping theory, an extension of the ideas in Salvucci and Anderson (1998a) in which analogical mapping is incorporated into a general problem-solving architecture. While other theories of analogy have successfully accounted for many aspects of human mapping behavior, they have yet to address rigorously the problem of integration with general problem solving. The path-mapping theory formalizes how to construct models of analogical problem solving that integrate mapping and other problem-solving skills. The theory proposes that people combine knowledge and skills for doing analogy with knowledge and skills specific to the content domain in which they are analogizing. We demonstrate that path-mapping models can quantitatively and qualitatively capture mapping behavior in a number of major phenomena explained by other theories. In addition, we show that path-mapping models can account for lower-level behavior in the execution of analogy not addressed by other theories.

Current Theories of Analogical Mapping

Numerous researchers have proposed theories that describe how analogical mapping takes place (Gentner, 1983, 1989; Hofstadter & Mitchell, 1994; Holyoak & Thagard, 1989; Hummel & Holyoak, 1997; Keane, Ledgeway, & Duff, 1994; Kokinov, 1998). These theories have made great strides in accounting for what we know of the mapping process. For instance, the theories have accounted for similarity effects between corresponding objects (Ross, 1987, 1989), pragmatic effects of focusing on particular concepts within analog domains (Spellman & Holyoak, 1996), and order effects of the adjacency of corresponding objects (Keane, Ledgeway, & Duff, 1994). The theories have focused primarily on the mapping process itself along with its interaction with other subprocesses of analogy. However, the theories have largely ignored issues of how analogy interacts with other aspects of problem solving — e.g., task-specific knowledge and skills — in a real-world task situation. Given that we now have theories of mapping that successfully capture a rich set of empirical findings, we can now begin to focus our efforts on how to incorporate these theories into more general theories of problem solving.

The fact that current theories of analogy and analogical mapping do not reside in more general problem-solving contexts limits them in two respects. First, the theories cannot incorporate certain types of task-specific knowledge and skills that might be needed in the course of performing a task. For instance, the performance of any analogical task requires the encoding of the source and target analogs and the generation of appropriate responses based on the resulting mappings. While most theories assume that these processes can be neatly separated from the analogy process, this assumption does not hold in general; for example, the story-mapping study in this paper demonstrates that people often interleave encoding and responding with the mapping process, incrementally mapping objects during each trial. Such results indicate that analogical mapping need not map all objects in a single process, but rather can be utilized incrementally according to task demands. As another example, tasks may place certain requirements on mapping that cannot be easily represented in current theories, such as task instructions to produce only one-to-one mappings.

The second limitation of current theories of analogical mapping, at least until very recently, is their reliance on coarse metrics of evaluation. These theories generally do not make direct predictions of real-world metrics such as latency or correctness, but rather make only rank-order correspondences between internal model computations and observable data. For example, Hummel and Holyoak (1997) compared the predictions of LISA with Ross's (1989) results by noting the correlation between the raw retrieval index generated by LISA and the proportions correct reported by Ross. While this argument demonstrates LISA's ability to model the Ross (1989) similarity phenomena, there remains further work in showing how such coarse metrics can be translated to predictions for real-world quantitative data. Again, existing theories are not inherently limited in this respect but have only recently begun to address this issue (e.g., Kubose, Holyoak, & Hummel, 1998).

The Path-Mapping Theory

This paper presents the path-mapping theory of analogical mapping that focuses on the integration of analogical mapping and general problem solving. The theory posits that people represent concepts declaratively as structures organizing objects in roles, map analogs by lower-level retrieval of analogous role paths, and coordinate mappings with higher-level organizational knowledge. The theory lies within a production-system architecture of cognition, ACT-R, which has captured psychological phenomena in numerous domains (Anderson & Lebiere, 1998). Thus, our approach to mapping differs from that of most researchers of analogy: Rather than working toward a general theory of cognition from a theory of analogy and mapping, we are setting a theory of mapping within the context of an existing general theory of cognition. Previous work in this vein (Anderson & Thompson, 1989; Salvucci & Anderson, 1998a) focused primarily on specific analogical tasks. In this paper we significantly extend these results into a much richer theory of analogical mapping that accounts for a wide variety of analogical tasks and phenomena.

The path-mapping theory addresses the two limitations of current theories discussed earlier. Situated in the ACT-R architecture, path-mapping models can utilize general problem-solving knowledge during analogical tasks. Path mapping represents a problem-solving skill available to models should they choose to utilize analogical mapping. Thus, these models can interleave mapping with other processes and can use the results of mapping in other cognitive computations as necessary for the task at hand. In addition, path-mapping models (like all ACT-R models) generate sequential simulation traces that describe the actions and action times during simulation. These traces allow for direct comparison between model predictions and human protocols with whatever real-world metrics are appropriate for the task.

We evaluate the path-mapping theory in two stages. First, we show that the theory can account for major qualitative and quantitative phenomena found in the existing analogy literature. Hummel and Holyoak (1997) provide a comprehensive list of criteria with which to evaluate a theory of analogical mapping. Their criteria include a number of empirical phenomena that a theory should account for. We show that the path-mapping theory captures these phenomena by presenting a number of path-mapping models for illustrative tasks and studies. Second, we demonstrate that the theory can capture how people interleave and coordinate the mapping process with other problem-solving processes during a task situation. Because existing data sets do not rigorously address this issue, we report the results of a novel story-mapping experiment that provides the low-level data needed to manifest the interaction between mapping and general problem solving. In the experiment, we recorded eye-movement and typing data as subjects determined the object mappings between related stories. These data provide a great deal of information about the step-by-step nature of analogical behavior that would otherwise be hidden from analysis. We then describe a model of the story-mapping task that accounts for subject behavior in terms of correctness, latency, eye-movement, and typing data.

Our exposition begins with an overview of the ACT-R cognitive architecture and a detailed description of the path-mapping theory. We then demonstrate how the path-mapping theory can account for many empirical findings that have been observed and modeled in the analogy literature. Next, we describe the story-mapping experiment and outline a model that successfully fits correctness, latency, encoding, and response data from the task. We conclude with a general discussion of various issues raised by the proposed theory.

The ACT-R Production-System Architecture

Production-system architectures provide for effective and convenient modeling of human behavior through production rules. A production rule defines a condition-action pairing such that, when the condition is satisfied, the given action is fired (i.e., executed). A model comprises a set of production rules that represent the control flow of the modeled behavior and a declarative memory store upon which the productions act. As productions fire, they produce a sequence of actions that can be compared to human actions in the same task. Existing production-system architectures such as ACT-R (Anderson & Lebiere, 1998), Soar (Newell, 1990), EPIC (Meyer & Kieras, 1997), and 3CAPS (Just & Carpenter, 1992) have been able to model human behavior in numerous tasks. (See Klahr, Langley, & Neches, 1987, for an overview and discussion of production-system architectures.)

The production-system architecture in which we base our theory of analogy is the ACT-R architecture (Anderson, 1993; Anderson & Lebiere, 1998). ACT-R has matured a great deal in recent years and has been used to model performance in a wide range of domains, including memory, choice, arithmetic, analogy, and scientific discovery (see Anderson & Lebiere, 1998). Given an ACT-R model that specifies declarative and procedural knowledge, a modeler can run simulations to obtain predictions of human performance for the model. ACT-R processes are stochastic, potentially resulting in different predictions across simulation runs. Because the

ACT-R theory is quite complex, we include in this exposition only those details that are relevant to the path-mapping theory. Interested readers are urged to consult Anderson and Lebiere (1998) for further information on ACT-R.

Declarative Knowledge

Declarative knowledge in ACT-R comprises **chunks** of knowledge that can be created and explicitly recalled. Each ACT-R chunk is an instantiation of a particular chunk type that defines what information can be contained by chunks of that type; specifically, the chunk type defines slots in which links to other chunks may reside. For instance, to model the knowledge 4+3=7, we may declare a chunk type *Addition-Fact* with slots *addend1*, *addend2*, and *sum*. We then can create chunks and model the knowledge as follows:

```
Four-Plus-Three
        isa             Addition-Fact
        addend1         Four
        addend2         Three
        sum             Seven
```

The *isa* description specifies the chunk type of which the chunk *Four-Plus-Three* is an instantiation. The three chunk slots (*addend1*, *addend2*, and *sum*) contain links to other chunks as needed (*Four*, *Three*, and *Seven*, respectively).

In addition to this symbolic representation, chunks have two sets of subsymbolic parameters relevant to the path-mapping theory. The first set of parameters, **chunk activations**, controls how quickly and successfully chunks can be retrieved. A chunk's activation is continually updated during simulation based on when the chunk was created, how often the chunk has been retrieved, and how long ago these retrievals occurred. Specifically, let us assume that a particular chunk i was created T time units in the past and has since been retrieved n times. The activation A_i for this chunk is defined as follows:[1]

$$A_i = nT^{-d} / (1-d) \qquad \text{(Equation 1)}$$

The decay factor d, which defaults to a value of 0.5, defines how quickly the activation from a particular retrieval drops off with time. Activation thus increases as the number of chunk retrievals (n) increases but decreases as the lifetime of the chunk (T) increases.

The second set of subsymbolic chunk parameters, **chunk mismatches**, represent the degree to which chunks are similar or dissimilar. The mismatch between a chunk and itself is set to 0, while its mismatch with other chunks can be any non-negative value. These mismatches, as we shall soon describe, are used in production firing to select similar chunks for retrieval and also to produce confusions in retrieval. ??? theory of where these come from? ??? At this time, ACT-R does not include a rigorous theory of similarity value settings and how they may be learned. Rather, the similarity values offer a simple and convenient representation for a possibly complex process of comparing objects and learning their similarities. ???

Procedural Knowledge

Procedural knowledge in ACT-R comprises **production rules**, or simply **productions**, that create and retrieve declarative chunks and send commands to modules that handle visual and motor processes. Production rules have two components: a left-hand side that describes the conditions under which the production can fire, and a right-hand side that describes the actions taken when the production fires. The left-hand side contains a specification of the current goal chunk; this goal chunk is the top-most chunk in the goal stack, which includes the current goal and its parent goals. The left-hand side can also contain a specification of a chunk to be retrieved from declarative memory. The right-hand side of a production rule specifies the actions executed when that production matches correctly and fires. These actions can include creating or modifying new chunks, pushing or popping goals from the goal stack, or sending commands to the visual or motor module.

The following example shows sample “pseudocode” for a production that retrieves the sum of two integers and types out the result:

```
Retrieve-Sum
        IF      the current goal is to add addend1 and addend2
                and the fact can be retrieved for addend1 and addend2 to get their sum
      THEN      type out the sum
                and pop the current goal
```

The left-hand side (the “If” clause) contains two chunk specifications. The first specification matches the current goal and assigns the values *addend1* and *addend2* to the values specified in the goal chunk. The second specification does the chunk retrieval, matching the chunk in declarative memory that holds the given addends and binding *sum* to their sum. The right-hand side (the “Then” clause) types out the computed sum and pops the current goal.

ACT-R has two sets of subsymbolic parameters for productions that are relevant to the path-mapping theory. The first set of parameters, **production match probabilities**, help decide which production to fire if multiple productions match on the left-hand side. In ACT-R, this decision actually arises from a set of parameters for each production, but it suffices to say that these parameters allow each competing production to fire with some probability. The second set of parameters, **production strengths**, describe how often the production has been used in the past. The next section describes how production strength scales the time needed to retrieve a chunk.

Simulation

The running of an ACT-R model simulation begins with the specification of the current goal (i.e., the top-most goal on the goal stack). At each step, the system finds the set of productions that could potentially fire, chooses which to fire, and fires the actions of the chosen production. The simulation ends when the model pops the top-most goal off the goal stack.

To fire a production that requires a chunk retrieval, ACT-R must determine whether a chunk can be retrieved and which chunk is retrieved. This process, known as **partial matching**, involves several steps. First, ACT-R finds all chunks of the specified type in memory and considers these chunks as candidates for retrieval. Second, ACT-R determines the **match score** M_{ip} of each candidate chunk *i* for production *p*, defined as

$$M_{ip} = A_i - D_{ip} \qquad \text{(Equation 2)}$$

where A_i is the chunk activation and D_{ip} is the degree of mismatch between the retrieval specification and the actual chunk, computed as the sum of chunk mismatches between the specified and actual chunks for each retrieval slot. Thus, the match score is simply the chunk activation for perfectly matching chunks but decreases as the similarity between specification and chunk decreases. ACT-R then adds some amount of Gaussian **activation noise** to each match score. Finally, ACT-R considers the chunk with the greatest computed match score: If this chunk's match score falls above the retrieval threshold of 0, the chunk is retrieved; otherwise, the retrieval fails.

Partial matching with noise allows ACT-R to account for two types of errors: errors of omission and errors of commission. For errors of omission, chunk retrievals can fail if the match score falls below the retrieval threshold. Returning to the addition example, an error of omission would fail to retrieve any addition fact for the two addends, effectively "forgetting" that fact. For errors of commission, ACT-R can potentially retrieve a chunk other than the intended one because of noise, causing the model to retrieve an incorrect answer. For instance, the Retrieve-Sum production, given the goal of retrieving the sum of *Four* and *Three*, might mistakenly retrieve a chunk *Four-Plus-Two* and type out the sum as *Six*. Both types of errors depend on the fact that match scores include some amount of noise during the retrieval process.

The latency of production firing depends on the latency of chunk retrieval and the action time for the production. If a chunk (other than the goal chunk) is retrieved in the left-hand side of the production, the latency T_{ip} of retrieving chunk *i* for production *p* is defined as

$$T_{ip} = e^{-(M_{ip}+S_p)} \qquad \text{(Equation 3)}$$

where M_{ip} is the match score for chunk *i* with the given specification and S_p is the production strength. Thus, as the match score and/or production strength increases, the latency of retrieval decreases to reflect faster use of that knowledge. In addition to retrieval time, each production firing incurs a default action time cost of 50 milliseconds. Specialized productions that perform some visual or motor task, such as encoding an on-screen item or typing a character, can be set to incur an additional time cost according to task demands.

Summary

In summary, we would like to emphasize three key aspects of the ACT-R architecture which will be essential to the path-mapping theory. First, ACT-R decomposes any large task into a number of production firings usually lasting only a few hundred milliseconds. This decomposition will be critical to the fine-grained analysis we offer of analogical mapping. Second, at all points, ACT-R’s performance is a function of the activation of the chunks it must retrieve. This aspect will allow us to model a number of effects of analog availability. Third, similarity through the partial matching process has a major impact on which chunks ACT-R retrieves. This feature will allow us to model similarity effects that permeate the analogy literature.

The Path-Mapping Theory and ACT-R Implementation

The path-mapping theory of analogical mapping represents a way of modeling analogical mapping within the ACT-R framework. The ACT-R architecture as a modeling tool is quite flexible; it allows for wide range of models for various task domains. While a cognitive architecture needs such flexibility for the sake of generality, it also requires constraints that limit the types of models one can write in the architecture. The path-mapping theory describes exactly these types of constraints. The theory helps explain analogical mapping by noting essential declarative representations and procedural knowledge that all models of mapping behavior share. It also describes how these essential elements can interact with other components of an ACT-R model to understand more fully the role of analogical mapping in general problem solving.

The path-mapping theory breaks down into three main components: representation, path mapping, and organization. Representation states how analog relations are represented as roles in declarative memory. Path mapping defines how the theory uses retrieval of analogous role paths to form analogical mappings between higher-order structures. Organization determines how models of analogical tasks can interleave analog retrievals with task-specific processes such as encoding and responding. We now detail these three components of the theory using the familiar solar-system/atom analogy (see Gentner, 1983; Keane, Ledgeway, & Duff, 1994) as an illustrative example.

Representation

In the path-mapping theory, analogs are represented as higher-order structures built up of three components: objects, relations, and roles. The first two components, **objects** and **relations**, serve the same purpose as in other theories of analogy (e.g., Gentner, 1983): Objects are the semantic primitives of the analogs, while relations link objects or relations together according to their function. For instance, Figure 1 shows the representation for the analogous solar-system and atom domains, with objects and relations displayed as ovals. The solar-system domain contains the two objects *ss-sun* and *ss-planet*, along with the three relations *ss-causes*, *ss-attracts*, and *ss-revolves*. Similarly, the atom domain contains the two objects *at-nucleus* and *at-electron* and the three relations *at-causes*, *at-attracts*, and *at-revolves*. (The *ss-* and *at-* prefixes specify tokens of their type for the solar-system and atom domains, respectively.)

[[Insert Figure 1 Approximately Here]]

The boxes in Figure 1 represent the third component of analog structure, **roles**, which serve to link objects and relations to form higher-order conceptual structures. Each role comprises five components:

- **parent** : a pointer to the parent relation
- **parent-type** : the semantic type of the parent relation
- **slot** : the relation slot that the object fills in the relation
- **child** : a pointer to the child object or relation
- **child-type** : the semantic type of the child object or relation

For instance, the box in the lower right of the source corresponds to the following role chunk:

ss-center

isa	role
parent	ss-revolves
parent-type	revolves
slot	center
child	ss-sun
child-type	sun

This chunk encodes the fact that the object *ss-sun* serves the *center* role in the relation *ss-revolves* — in other words, the sun is the center around which something revolves. By linking objects and relations using such roles, we can form higher-order structures of arbitrary complexity. Figure 1 includes the roles, shown as rectangles, necessary to link the objects and relations in the solar-system and atom domains. The roles *ss-causes* and *at-causes* illustrate how roles can link not only relations to objects but also higher-order relations to lower-order ones. Note that this model decomposes a relationship like a planet revolving around the sun into a set of separate chunks encoding the role of each object (planet, sun) in the relationship. This decomposition is critical to accounting for a number of phenomena, including the ability to map relations with different numbers of arguments.

The following section on path mapping will shed more light on the motivation behind many details in this representation. Generally speaking, this representation for analogs makes sense in a number of ways. The presence of parent-type, slot, and child-type information in the same role allows all three to interact in the mapping process, which helps to account for similarity-based confusions between possible mappings. The inclusion of the parent and child pointers in the roles allows the model to maintain structural constraints during mapping. Overall, the representation closely resembles that in the LISA model (Hummel & Holyoak, 1997), where objects are linked to propositions by means of mediating subpropositions that describe the role of the object within the proposition. The representation is also reminiscent of "positional" representations used in list-memory models and similar ACT-R models (Anderson, Bothell, Lebiere, & Matessa, 1998) that include pointers to higher-level chunk structures and the position in the structures with the representation of list elements.

Path Mapping

Path mapping acts as the fundamental and most basic process of analogical mapping. Path mapping maps a single source object to a single target object by finding analogous paths between the objects and their highest-order parent relations, or **root relations**. This process operates in two stages: first, it determines a source path from the source object to its root relation; and second, it maps the relations and objects along this path to analogous relations and objects in the target. By mapping objects one at a time, path mapping provides an incremental mechanism that allows models of analogy to incorporate and interleave analogical mapping and other general problem solving skills. In addition, it provides sensitivity to the structure of the analog representations while allowing flexibility in mapping between non-identical structures.

The first stage of path mapping involves determining the path from the given source object to its root relation. Given the source object, we begin by retrieving a role for the object in the source domain. We then extract the parent relation from the retrieved role and iterate the process with the parent relation as the object. This process recursively retrieves roles for parent relations until no such role can be retrieved — that is, when the process reaches the root relation. For instance, Figure 1 shows a sample source path (in heavy lines) for the object *ss-planet* in the source domain. The model retrieves the role *ss-revolver* for *ss-planet*, retrieves its parent relation *ss-revolves*, retrieves its role *ss-effect*, and finally retrieves the root relation *ss-causes*. Another valid path would rise from *ss-planet* through *ss-attracts* to *ss-causes*; however, path mapping would choose only one of these two possible paths in a single attempt to map *ss-planet*. In identifying such a path, the model effectively "understands" the role of the object in the source domain.

The second stage of path mapping maps each relation and object along the source path to its analog in the target. Starting with the source root role (i.e., the role with the source root relation as parent), path mapping finds the most similar target role and maps the source role's parent and child to the target role's parent and child (respectively). Similarity between the source and target role is determined based on the similarities between the parent-type, slot, and child-type components of the roles. Then, path mapping proceeds down the source path mapping each child along the path to its analog, ending with the final mapping between source object and target object. For example, to map *ss-planet* to its analog in Figure 1, we map the source path derived earlier to an analogous target path (shown as heavy lines in the target domain). This process begins by retrieving the target role that is most similar to the source root role *ss-effect*; in this case, path mapping would retrieve *at-effect* as the target role that is most similar to *ss-effect*. With this retrieval, path mapping creates a mapping between the roles' parents and children; thus, it forms the mappings *ss-causes* to *at-causes* and *ss-revolves* to *at-revolves*. Path mapping then continues this process recursively, retrieving a target role that is a child of *at-revolves* and is most similar to *ss-revolver* — namely, *at-revolver*. Finally, path mapping produces the mapping *ss-planet* to *at-electron* and returns *at-electron* as the analog for *ss-planet*.

So far we have discussed path mapping only in the unconstrained case where any source path or target path may be retrieved. However, there are situations — as we describe in later sections — where path mapping may be constrained to a particular source or target role. The source role constraint specifies which source role should be used when retrieving the source path; for instance, in Figure 1, path mapping may be constrained to use the *ss-attracted* role for *ss-planet*. This constraint allows people to map a source object for a single particular role rather than for any retrieved role. The target role constraint specifies which target role should be tried when retrieving the analogous target path. This constraint allows people to restrict mapping to a particular target role, thus evaluating the target role as a possible analog: If the mapping succeeds, path mapping returns a target object that fills the target role, otherwise path mapping returns failure.

Path mapping thus maps a single object from the source analog to a single object in the target analog. In doing so, it records the mappings of both objects and relations along the mapped path. Path mapping may utilize these recorded mappings when mapping other objects in the domains, thus using the structural constraints of earlier mappings. In the Figure 1 example discussed above, three mappings were produced: *ss-causes* to *at-causes*, *ss-revolves* to *at-revolves*, and *ss-planet* to *at-electron*. If we then attempt to map *ss-sun*, and the *ss-center* role was included in the source path, path mapping would recall the mapping from *ss-revolves* to *at-revolves* and would not proceed further in mapping the higher-level relations. The reliance on previous mapping results thus provides structural constraints when mapping multiple objects between domains.

In ACT-R, path mapping is implemented as a set of production rules that operate over a specific *map-object* subgoal. The input to the path-mapping productions comprise the source object to be mapped and, optionally, a specified source role and/or target role constraint. Upon completion of mapping, the productions return either an analogous target object or a *failure* symbol.

The productions appear in pseudocode in Table 1.[2] Retrieve-Previous-Mapping immediately returns the analogous target object if it can retrieve a previously-computed mapping for the given source object. Retrieve-Source-Role retrieves a source role for the given source object. If no source role can be retrieved, this means that the source object is at the root of the source path, and Reached-Source-Path-Root notes that a complete source path has been found. Retrieve-Components extracts the components of the retrieved source role — namely, the source relation, parent type, slot, and child type specified in the role. Map-Source-Relation recursively maps the source relation to an analogous target relation — that is, it pushes a new subgoal in which the source relation becomes the source object and is mapped to its appropriate analog. The final two productions implement the second phase of path mapping: each retrieves a target role that is analogous to the source role using the source role components. At the root, Retrieve-Analog-At-Root retrieves an analogous target role and returns its target object. Below the root, Retrieve-Analog-Below-Root retrieves an analogous target role for the mapped target relation and returns its target object.

[[Insert Table 1 Approximately Here]]

To illustrate better the control flow of production firings during analogical mapping, Table 2 shows a sample trace of these productions mapping the *ss-planet* object in the Figure 1 example. The table displays the cycle number, the production that fires at each cycle, and a description of what occurs; note that the descriptions are indented to represent the goal depth, such that all lines at the same indentation represent production firings for a particular goal. The model begins with the goal of mapping *ss-planet*. In cycle 1, Retrieve-Source-Role retrieves the source role *ss-revolver* for *ss-planet* and sets it as the source role in the current goal. In cycle 2, Retrieve-Components extracts the source relation, parent-type, slot, and child-type components from the source role and set these values in the current goal. In cycle 3, Map-Source-Relation pushes a new subgoal to map the source relation *ss-revolves* to its analogous target relation. In cycles 4 and 5, the model fires Retrieve-Source-Role and Retrieve-Components to retrieve the source role *ss-effect* and the other components for *ss-revolves*; note that the descriptions in these cycles are further indented to indicate the new subgoal. In cycle 6, Map-Source-Relation pushes a new subgoal to map *ss-causes*. Because *ss-causes* is a root relation (i.e., it has no relations above it), Reached-Source-Path-Root fires in cycle 7 and notes that the root has been reached. In cycle 8, Retrieve-Analog-At-Root retrieves *at-effect* as an analogous target role for *ss-effect* and records the mappings for the relations and objects. Finally, in cycle 9, Retrieve-Analog-Below-Root retrieves *at-revolver* as an analogous target role for *ss-revolver* and records the object mapping.

[[Insert Table 2 Approximately Here]]

Two points regarding this example require some elaboration. First, the example demonstrates the trace for mapping only a single object. If we were to map *ss-sun* after mapping *ss-planet*, we may find that the mappings generated for *ss-planet* would constrain the mappings for *ss-sun*. For instance, if the *ss-center* role were retrieved for *ss-sun*, path mapping would note the previous mapping from *ss-revolves* to *at-revolves* and would not proceed further up the source path (i.e., would not re-map *ss-causes*). Second, the example trace shows only one possible trace for mapping *ss-planet*. The model could select *ss-attracted* as the source role for *ss-planet*, or may incur a retrieval failure at some point in the trace if activation noise lowers the chunk's match score below ACT-R's retrieval threshold.

Organization

Path mapping acts as the fundamental low-level mechanism for analogical mapping. However, analogical behavior requires more than simple mappings of single analogous objects; people must know how to encode analogs, when to use path mapping, and how to coordinate mappings to solve the task at hand. This knowledge can be thought of as higher-level organizational knowledge that decides when and how to employ the lower-level path-mapping mechanism. For instance, in the solar-system/atom example, organizational knowledge would include any task-specific knowledge or skills needed for the particular task context. Suppose that subjects are provided with paper-and-pencil tests such that they must map the sun and planet in the solar-system domain to analogous objects in the atom domain. The organizational knowledge needed for such a task might specify, for instance, how subjects read the given test materials, how and when they perform the mapping for each object, and how and when they write down their responses. Thus, organizational knowledge describes the way in which people interleave and interface path mapping and other cognitive skills for a particular task.

The separation of higher-level organizational knowledge and the lower-level path-mapping mechanism is an important one. While path mapping acts on a very general representation that transfers naturally across domains, organizational knowledge can be task-specific and could vary widely across domains. Thus, models of analogical mapping for two different tasks would contain the identical path-mapping mechanism but might contain very different mechanisms for organizational skills. Even a single task model might utilize path mapping in various ways, helping to account for variability in analogical behavior as has been observed in several studies (e.g., Grudin, 1980; Salvucci & Anderson, 1998a; VanLehn, 1998).

Issues

Origins of Mapping Phenomena

The path-mapping theory and its implementation in ACT-R provide a number of interesting clues to the origins of various observed mapping phenomena. One such phenomenon, the effect of systematicity (Gentner, 1983), is incorporated directly in the path-mapping component of the theory. The systematicity principle states that relations that participate in common systems of relations are more likely to map than isolated relations. By tracing the source path up to the root role by mapping the root role first, and by constraining other mappings to be consistent with this mapping, path mapping attempts to uncover the most systematic mapping possible.

path mapping attempts to map the highest-order relation possible, thus preserving systematicity. Thus, the path-mapping theory directly incorporates the systematicity principle as an assumption of how people perform analogical mapping.

While the path-mapping theory directly incorporates the systematicity principle, two other mapping phenomena arise as emergent phenomena from architectural aspects of ACT-R. First, the path-mapping model predicts pragmatic effects where emphasized concepts are more readily mapped than unemphasized ones (Spellman & Holyoak, 1996). When a task emphasizes a particular role, the chunks involved are repeatedly used and thus become more active than other chunks. During path mapping, this increased activation boosts the match score of emphasized chunks and results in a greater chance that they will be retrieved in mapping. These pragmatic effects can affect both the first stage of path mapping, in retrieval of a source role, and the second stage, in retrieval of an analogous target role. Second, the path-mapping model predicts similarity effects where surface similarity can interact with role correspondence (Ross 1987, 1989). The path-mapping productions, through partial matching (see Equation 2), use parent-type, slot, and child-type values to determine the best analogous target role. When similar objects play corresponding roles, the similarity between correct analogs is high, making it likely that the model will retrieve the correct analog. However, when similar objects play non-corresponding roles, the similarity between correct analogs is low, decreasing the chance that the correct role will be retrieved. The path-mapping model thus predicts possible confusion between roles with similar slot values and roles with similar child-type values. The path-mapping implementation, then, produces emergent effects of pragmatic emphasis and similarity as a result of its instantiation in the ACT-R architecture.

While the path-mapping component of the theory helps explain systematicity, similarity, and pragmatic effects, the organization component helps explain several other important empirical phenomena such as order effects and the generation of non-isomorphic mappings. Order effects come from the manner in which organizational knowledge encodes and processes the given analogs. Non-isomorphic mappings arise from computational constraints imposed by organizational knowledge. The next section provides a number of illustrative path-mapping models that more fully describe how the path-mapping theory can account for these phenomena.

Computational Complexity

Path mapping is an efficient, “greedy” process: It selects a single path in the source analog and maps it to a best-matching path in the target analog. In doing so, it maps each object and relation on the source path to their corresponding objects and relations on the target path. If, in mapping another object, an already-mapped relation is reached, path mapping simply uses its previous mapping and does not consider any higher-order relations above the mapped relation. To map every source object to a target object, path mapping maps each object and relation at most once, thus running in time linear to the total number of objects and relations.

Path mapping’s greedy approach to analogical mapping is quite different from early models of mapping that take a more global approach (e.g., Falkenhainer, Forbus, & Gentner, 1989; Holyoak & Thagard, 1989) and more reminiscent of recent models that take a more local or incremental approach (e.g., Forbus, Ferguson, & Gentner, 1994; Keane & Brayshaw, 1988). The rationale for these efficient approaches arises from the fact that any attempt to exhaustively search for a mapping takes time that is exponential in the size of the analogs and is thus implausible for complex domains. However, while other theories include mechanisms for backtracking and reevaluation to form more optimal mappings, the path-mapping theory provides a simple path-mapping module with the ability to perform more complex operations through the organizational model. The attribute-mapping model presented in the next section illustrates how organizational knowledge can utilize backtracking to evaluate mappings and rebuild more optimal mappings.

Parameter Consistency

A major goal of the path-mapping theory is to constrain the space of possible ACT-R models of analogical mapping. Along this line, the theory should also constrain the space of possible parameter values for these path-mapping models. For all models in this paper, we strove to achieve parameter consistency in two ways: identifying important mapping-related parameters and estimating single values for them across all models, and presetting other parameters to reasonable values for all models. We now describe the setting of these parameters, as summarized in Table 3.

[[Insert Table 3 Approximately Here]]

The mismatch parameters define the chunk mismatches for relation and object type chunks used in calculating degree of mismatch (D_{ip} in Equation 2). All models presented in this paper use the same two mismatch parameters — one for similar chunks and one for dissimilar chunks. For instance, the chunks *revolves* and *rotates* are similar whereas *revolves* and *walks* are dissimilar. The chunk mismatches between similar and dissimilar chunks help to account for correct matching and confusion in chunk retrieval. We estimated the similar-chunk mismatch to be 0.1 and the dissimilar-chunk mismatch to be 1.7.

The path-mapping models also contain three preset parameters — that is, parameters not estimated but rather preset to reasonable values and kept constant across all models. The first two parameters, activation noise and activation learning decay rate (*d* in Equation 1), were set to values that have been found to work well in a number of previous ACT-R models (see Anderson & Lebiere, 1998): activation noise was set to 0.5 and the activation learning decay rate was set to 0.5. In addition, we preset the number of references (*n* in Equation 1) given to all declarative chunks at the start of simulation to 50, thus modeling practice given to these chunks before analogizing.

Illustrative Path-Mapping Models

We now demonstrate how the path-mapping theory successfully accounts for major qualitative and quantitative phenomena found in the existing analogy literature. Hummel and Holyoak (1997) provide a comprehensive list of phenomena with which we can evaluate the path-mapping theory. Table 4 includes their phenomena (1-7) along with three of our own (8-10). Isomorphism states that people utilize structural and relational cues to facilitate mapping when corresponding objects are dissimilar (Gick & Holyoak, 1980) or when non-corresponding objects are similar (Ross, 1987, 1989). Semantic similarity can ease mapping when between corresponding objects but can hinder mapping when between non-corresponding objects (Gentner & Toupin, 1986; Ross, 1987, 1989). Pragmatic centrality implies that relations emphasized or deemed important are more likely to participate in mapping (Spellman & Holyoak, 1996). Studies have also shown that mappings for a particular analogy need not be unique (Spellman & Holyoak, 1996) and that the order of analog relations can effect the time needed to map analogs (Keane, Ledgeway, & Duff, 1994). In “unnatural” analogy problems such as the attribute-mapping problem (Holyoak & Thagard, 1989), people often have difficulty finding plausible mappings and need to backtrack through multiple possible solutions (Keane, Ledgeway, & Duff, 1994). The ability to map relations with different number of arguments has also been informally viewed as a desirable trait for theories of mapping. People can sometimes generate non-isomorphic mappings with a strong preference for many-to-one over one-to-many mappings (Spellman & Holyoak, 1996). Finally, any model of analogical mapping should be able to handle complex analogies with reasonable computational cost (Veale & Keane, 1997). The remainder of this section presents models that illustrate how the path-mapping theory accounts for these phenomena.[3]

[[Insert Table 4 Approximately Here]]

The Probability-Problem Model

Ross (1989) studied in great detail the effects of semantic similarity on analogical mapping. In a series of experiments, subjects studied sample probability problems and solved test problems based on these sample problems. Ross measured subjects' performance on this task as the proportion of problems in which subjects correctly mapped corresponding objects from the sample problem to the test problem. His results illustrate that semantic similarity can positively influence analogical mapping when similar objects correspond and can negatively influence mapping when similar objects do not correspond.

Ross ran five experimental conditions based on the similarity of story lines and corresponding objects, categorized by the notation {+/0}/{+/–/0}. The first symbol in this notation represents the similarity of story lines, where + indicates similar story lines and 0 represents dissimilar story lines. The second symbol represents the similarity of corresponding objects, where + indicates that similar objects correspond, – indicates that similar objects do not correspond, and 0 indicates that similarities are neutral with respect to correspondences. Table 5 illustrates the relations for a sample problem and five test problems in the conditions +/+, +/–, 0/+, 0/–, and 0/0. The probability problems could be solved with an equation using the variable n which represented the number of objects being assigned; for example, in the sample problem, the number of cars corresponded to the variable n.

[[Insert Table 5 Approximately Here]]

Table 5 shows the correctness results of these experiments as the proportion of subjects who mapped the variable *n* to the correct corresponding test problem object — that is, mapped the source object *cars* to the target object being assigned. Subjects' performance was highest when similar objects between the sample and test problems corresponded (+/+) and was lowest when similar objects did not correspond (+/–). When story lines differed, the effect between conditions with corresponding (0/+) and non-corresponding (0/–) objects decreased slightly but remained significant. (Note: The value for the 0/– condition is the average of two values reported in Ross, 1989, for separate experiments.) In the absence of any relation between sample and test objects (0/0), performance fell in the middle of all conditions.

We created a path-mapping model that performs the mapping between sample and test problems similar to the model by Hummel and Holyoak (1997). The model's representation of the analogs uses a straightforward application of the role chunk representation on the relations in Table 5. The model comprises only the productions in the path-mapping module — that is, the productions in Table 1. In simulation, the model maps the source object *cars* to a corresponding target object. First, path mapping retrieves the source role of the object *cars* in the sample problem. Second, path mapping retrieves an analogous target role in the current test problem. Correctness was computed for simulations just as for subjects — namely, whether they mapped the source object *cars* to the target object being assigned.

In parameter fitting, the model used the estimated and preset parameter values in Table 3. In addition, the model contained a single estimated task-specific parameter: the probability of a miscellaneous error when solving the problem. The Ross (1989) data show a high degree of error (.40) for even the most straightforward condition (+/+). Although the model could account for these errors with artificially high activation noise, we believe that many of these errors arose from parts of the problem not modeled here — for instance, not making use of the formula or incorrectly instantiating a variable (as mentioned in Ross, 1987). We estimated the probability of a miscellaneous error to be .36.

The results for 1000 simulations of the model are shown in Table 5 and correspond well to the data, *R*=.93. The model captures the major trends in the data — namely, the large effect between +/+ and +/–, the smaller but significant effect between 0/+ and 0/–, and the mean performance for 0/0. The differences among the conditions are largely produced by the frequency with which the model retrieves the wrong role chunk to map *cars*. In all conditions the role of *cars* (*from*) predisposes ACT-R to retrieve the correct role chunk, but the object similarity to *cars* creates an activation boost that can push the choice somewhat in either the correct or incorrect direction. In the +/+ and 0/+ conditions, the activation boost from the similar objects in corresponding roles leads the model to choose these roles almost always; in the +/– and 0/– conditions, the activation boost from similar objects in non-corresponding roles leads the model to choose these roles with some frequency. The difference between the +/+ and +/– conditions is slightly greater than the difference between the 0/+ and 0/– conditions because the former conditions have identical objects that increase the similarity effect. In the 0/0 condition, both the corresponding and non-corresponding roles carry the same activation boost, and thus the model chooses the incorrect role with some frequency. Overall, the model's predictions correlate well with the observed results and support the ability of the path-mapping theory to account for Phenomena 1 and 2.

The Soap-Opera Model

Spellman and Holyoak (1996) examined the effects of pragmatic centrality on analogical mapping. They found that focusing subjects' attention on particular relations in the source and target analogs can affect the mappings that subjects produce from source to target. We now discuss Spellman and Holyoak's soap-opera experiment (Experiment 3) along with a path-mapping model that predicts the observed data.

In the soap-opera experiment, subjects read two similar soap-opera plots and compared the plots under the guise of determining whether one writer had plagiarized the other. The plots centered on the romantic, professional, and cheating relations schematized in Table 6. Subjects performed two primary tasks during the experiment: a plot-extension task and a mapping task. In the plot-extension task, subjects extended a given target story using an analogous source story as a guide. The source story emphasized a particular type of relation (romantic or professional) and thus subjects were expected to write continuations focusing on the emphasized relations. In the mapping task, subjects mapped each character from the source relations to a character from the target relations.

[[Insert Table 6 Approximately Here]]

Spellman and Holyoak analyzed the experimental results in terms of how subjects mapped *Peter* and *Mary*. Both *Peter* and *Mary* could justifiably map to any character in the target analog, but pragmatic emphasis and the cheating relations made some mappings more likely than others. For instance, although *Peter* could map to any character, an emphasis on the *bosses* relation would lead to more likely mapping to *Nancy* or *David*; when adding the additional constraint of the cheating relations, *Peter* would most likely map to *Nancy*. Based on these observations, Spellman and Holyoak separated mappings as to whether they were consistent or inconsistent with pragmatic emphasis (CP or IP), consistent or inconsistent with the cheating relations (CC or IC), or some other category (Other). For instance, if we assume that the *bosses* relations are given pragmatic emphasis, the following pairs list the various categories and characters to whom *Peter* and *Mary* would map: CP-CC, *Nancy* and *John*; CP-IC, *David* and *Lisa*; IP-CC, *Lisa* and *David*; IP-IC, *John* and *Nancy*; other, any other pairing.

Figure 2 graphs Spellman and Holyoak's experimental results for the plot-extension and mapping tasks, where each bar indicates the percentage of subjects in the various categories. In the plot-extension task, pragmatic emphasis had a large effect on how subjects used *Peter* and *Mary* because subjects were instructed to use the particular emphasized relations in extending the plots. The cheating relations, however, had no significant effect on subject behavior. In the mapping task, where subjects had no explicit guidance toward the emphasized relations, both pragmatic emphasis and the cheating relations had a significant effect on behavior. Thus, in the absence of explicit guidance, pragmatic focus can affect mapping but other factors — namely the presence of unemphasized relations — can also intrude.

[[Insert Figure 2 Approximately Here]]

Our model for this task is very similar to that for Ross's task. The representation uses role chunks for each object in the analog relations in Table 6. The production system comprises the path-mapping productions and five additional productions that iteratively map *Peter* and *Mary*. In simulation, the model first retrieves a source role for *Peter*: In the plot-extension task, this retrieval is constrained to the emphasized relation, modeling the explicit guidance given to subjects; in the mapping task, this retrieval is unconstrained and can retrieve any of *Peter*'s source roles. Next, the model maps *Peter* to an analogous target object with the retrieved source role as a constraint. The model then repeats this process for *Mary*, constraining path mapping to a source role for the same source relation used in mapping *Peter*. Thus, the model maintains structural consistency in two ways. First, if possible, the model uses the same source relation to retrieve source roles for *Peter* and *Mary*; for instance, if the model retrieves *Peter*'s source role in the *loves* relation, it also retrieves *Mary*'s source role in the *loves* relation. Second, because path mapping records the results of mappings in declarative memory, the mappings formed when mapping *Peter* may constrain those when mapping *Mary*; for example, if *Peter* maps to *John* and the source *loves* relation maps to the first target *loves* relation (which contains *John*), the relation mapping constrains *Mary* to map to *Nancy*.

For parameter fitting, as in all models, the soap-opera model used the shared parameters in Table 3. In addition, the model contained a parameter that specified the number of additional references given to chunks that receive pragmatic emphasis according to the task condition. This increased reference count gave emphasized chunks a higher activation than unemphasized chunks, increasing their likelihood of being used in mapping. We estimated that emphasized chunks receive 50 additional references (added to the 50 standard references in Table 3).

The results for 1000 simulations are presented in Figure 2. The model provides a close fit to the data, *R*=.98. In the plot-extension task, the model produces the large main effect of the pragmatic manipulation on the task. Since source role retrieval is constrained to emphasized relations via explicit instructions, the model usually retrieves the emphasized roles and uses these roles to retrieve analogous target roles. Occasionally, because of activation noise, the model produces a mapping other than the one implied by the pragmatic manipulation, as reflected in the other category. The model, like subjects, exhibits no effect of the cheating relations. In the mapping task, the model sometimes uses non-pragmatic source roles for analogy. However, the model still produces a pragmatic effect due to the increased activation of pragmatic roles that makes these roles more likely to be retrieved. The model also predicts the cheating effect: Because the model sometimes chooses a cheating role as the source role, mappings using this cheating role are more likely than other mappings. Again, because of activation noise, the model also predicts mapping pairs that do not conform to any of the four standard categories. Thus, by modeling pragmatic emphasis as increased activation of emphasized roles, the model captures observed behavior in both the plot-extension and mapping tasks. The model demonstrates the ability of the path-mapping theory to account for Phenomena 1, 3, and 4.

The Attribute-Mapping Model

Keane, Ledgeway, and Duff (1994) showed that the ordering of analog relations can affect the ease of mapping in the attribute-mapping task. The attribute-mapping task has subjects determine the mapping between boys and dogs for story relations like the following:

Bill is intelligent.	Fido is clever.
Bill is tall.	Blackie is friendly.
Tom is timid.	Blackie is frisky.

Tom is tall.	Rover is clever.
Steve is intelligent.	Rover is friendly.

The stories contain one singleton relation — the only relation for a particular boy or dog — and two pairs of relations for the other boys or dogs. The correct mapping is defined by the constraints given in the adjectives used to describe the boys and dogs; for the above example, the correct mappings would comprise *Bill* to *Rover*, *Tom* to *Blackie*, and *Steve* to *Fido*. Subjects often have difficulty with the task, requiring several minutes on average to complete the task.

In two experiments, Keane et al. examined the effects of order and attribute similarity on subject task time. In the order experiment, they found that singleton-first stories where both singletons appeared first required significantly less time than singleton-last stories where singletons appeared in non-adjacent positions. In the similarity experiment, they observed that mapping difficulty decreased as the number of similar attributes (e.g., *intelligent* and *clever* in the above example) increased — namely, from the none-similar to one-similar to all-similar conditions. Figure 3 shows the average task times for each condition; because the singleton-last and none-similar conditions used essentially the same materials, these task times have been averaged into one condition.

[[Insert Figure 3 Approximately Here]]

We created a path-mapping model to simulate performance in the attribute-mapping task. The model's primary subgoal tries to create mappings for the current source relation starting with the first relation. For each position, the model runs through two stages: a *retrieve-role* stage and an *iterate-role* stage. In the *retrieve-role* stage, the model retrieves the source role at the current position and attempts to map the source object using this source role as a constraint. If successful, the model records a mapping between the source and target relations and objects and tries to create mappings for the next source position; if unsuccessful, the model moves on to the *iterate-role* stage. In the *iterate-role* stage, the model begins with the first target role and tries to map the current source role to this target role: It maps the relations and objects and tries to map the next source role. When a mapping attempt fails, the mapping is deleted and the model moves to the next target role; the model recognizes a mapping as incorrect when it is inconsistent with currently-recorded mappings. When mapping attempts fail for all target role positions, the model backtracks to previous source roles and tries other target mappings. The recording and deleting of mappings is assumed to occur on written materials (as subjects had in the task). Thus, the model implements a recursive backtracking strategy that tries possible mappings until the correct mapping is found.

The model predictions for the task conditions, included in Figure 3, were run using only the shared parameters in Table 3 — no additional parameters were included. The model produces an excellent correlation with the empirical data, *R*=.97. The model predicts that the singleton-first condition, where singletons appear first in both analog, gives the fastest times due to the fact that roles are analyzed in a top-down fashion. The model also predicts relatively fast times in the all-similar and one-similar conditions: Because the model attempts to retrieve analogous target roles before iterating, it favors conditions with more similarity between corresponding roles. The model predicts the longest times for the singleton-last/none-similar condition, where the singletons are out of order and there is no similarity to guide the mapping process. Although we could further improve the quantitative fit by adding a parameter that specifies time needed to encode the analogs or write the results, such an effort would show little about the path-mapping theory and is better left for later work. Overall, these results demonstrate that the path-mapping theory can account for Phenomena 2, 5, and 6. In particular, this model emphasizes the importance of the theory's reliance on similarity to guide mapping and the incremental nature of the mapping process.

The "Sharing" Model

Some researchers (e.g., Hummel & Holyoak, 1997) have stated an informal qualitative goal that theories of analogy should allow for mappings between relations with different numbers of arguments. All existing theories of analogy, with the exception of that embodied by LISA (Hummel & Holyoak, 1997), require that mappings follow an *n*-ary restriction — that is, only map between relations with the same number of arguments. While people may typically follow this constraint, it seems reasonable to believe that this need not be the case. For instance, consider the source relation *shared (Jim, candy, Cathy)* and the target relation *shared (Cindy, toys)*. In this example, the sharing relation need not provide all of its possible arguments for a particular instance of the relation. However, we would still expect humans to map *Jim* to *Cindy* and *candy* to *toys*.

We created a path-mapping model that encodes the above analogs and generates a mapping for *candy*. The model contains only the path-mapping productions in Table 1. Because the model represents each object as a role chunk, the number of arguments for each relation does not affect its ability to map successfully. By encoding the fact that *Jim* and *Cindy* are the actors in the *shared* relation and that *candy* and *toys* are the objects of the relation, the model easily maps *candy* to the corresponding target object *toys*. Thus, the path-mapping theory can account for Phenomena 7.

The Country-Mapping Model

In the analogy process, people typically generate one-to-one mappings — that is, mappings where each object in the source analog corresponds with exactly one object in the target analog. Some theories of analogy (e.g., Gentner, 1983) require that mappings be one-to-one. However, Spellman and Holyoak (1996) described results that show that people sometimes break the rigid constraints of one-to-one mappings. We show that the path-mapping theory can account for these results.

In their Experiment 2, Spellman and Holyoak gave subjects two stories involving countries on different planets and asked subjects to map countries on one planet to those on the other. The story relations can be summarized as follows:

Story 1	Story 2
richer (Afflu, Barebrute)	richer (Grainwell, Hungerall)
stronger (Barebrute, Compak)	stronger (Millpower, Mightless)

The relations include an ambiguous mapping — namely, the mapping of *Barebrute* to either *Hungerall* or *Millpower*. Subjects were divided into two conditions: In the 1→2 condition, subjects mapped objects from story 1 to those in story 2; in the 2→1 condition, subjects mapped objects from story 2 to story 1. In both conditions, subjects had the option of including any, all, or no objects in their mapping, thus allowing the possibility of a one-to-one, one-to-many, or many-to-one mapping if so desired. Spellman and Holyoak's analysis states that one-to-many mappings were "rarely produced" (less than 2% of subjects) while many-to-one mappings were "frequently produced" (more than 30% of subjects).

The path-mapping theory posits that the distinction between one-to-one, one-to-many, and many-to-one mappings arises in the organizational productions specific to the task. We created a path-mapping model that demonstrates how such organizational knowledge can be realized. The model, like subjects, is instructed either to map objects from story 1 to story 2 (the 1→2 condition) or to map objects from story 2 to story 1 (the 2→1 condition). The model simply iterates through each object in the source story and maps the object to its analog in the target story. If the model finds a mapping from a source object to a target object that has already been mapped, it decides whether or not to write down the mapping with some probability. This decision serves to model subject differences in the task: Because subjects were not instructed whether they should or should not produce one-to-one mappings, we suspect that some subjects assumed a one-to-one constraint whereas others did not.

The country-mapping model reproduces the behavior of subjects in that it can generate many-to-one mappings but not one-to-many mappings. In the 2→1 condition, the model maps each story 2 object to a story 1 object. Because it maps each story 2 object exactly once, the model maps both *Hungerall* and *Millpower* to *Barebrute*. Because some subjects maintained a one-to-one mapping restriction, we also allow the model not to record a second mapping for a particular target with some probability. In the 1→2 condition, the model maps each story 1 object to a story 2 object. Like in the 2→1 condition, it maps each story 1 object exactly once, thus the model never has the opportunity to generate a one-to-many mapping: The model simply attempts to find an analog for *Barebrute*, chooses either *Hungerall* or *Millpower*, and records this mapping. Thus, the presence of many-to-one mappings and lack of one-to-many mappings arises from the fact that subjects map each source objects to a target object exactly once. This model demonstrates how the path-mapping theory accounts for Phenomena 8 and 9.

The Karla-the-Hawk Model

Thus far, the models we have presented have involved relatively small analogs with few objects and relations. However, the path-mapping theory easily scales up to even complex domains with large, highly-associative analogs. To illustrate this point, we examined two stories taken from an experiment examining analogical access (Gentner & Landers, 1985). In one story, a hawk named Karla is attacked by a hunter but comes to befriend the hunter by giving him some of its feathers. In an analogous story, the country of Zerdia is attacked by the country of Gagrach but eventually befriends Gagrach by sharing its computers. The two stories are fairly complex and require a representation with a number of objects and relations.

We created a model of the Karla-the-Hawk stories using a representation based on that of Falkenhainer, Forbus, & Gentner (1989). This representation includes four objects, 18 relations, and 36 roles for each story analog. The model runs through the four objects and maps each object to a corresponding target object. Because mapping takes time linear to the number of objects and relations, the model produces the mappings very quickly: the simulation itself takes less than a second to run, and it predicts that people require approximately 23 s to map all objects correctly. This model shows that the path-mapping theory accounts for Phenomenon 10.

Summary

We have described a number of illustrative models that show how the path-mapping theory accounts for important phenomena in analogical mapping. While other theories of analogy have also proven successful at modeling many of these phenomena, they have relied on coarse metrics of comparison between model predictions and empirical data. We have evaluated the path-mapping theory by comparing model predictions directly to real-world metrics when possible and, as summarized in Table 4, the theory provides good qualitative and quantitative fits to the data. The path-mapping theory is, to our knowledge, the first theory of analogical mapping to account for this rich body of phenomena in terms of these real-world metrics.

The Story-Mapping Study

The previous section evaluated the path-mapping theory using existing data sets that illustrate particular aspects of the analogical mapping process. While these data sets provide a rich basis for understanding the mapping process itself, they provide little insight into the integration of mapping and other general problem-solving processes; for instance, they completely ignore issues of encoding and responding during a task situation. Accounting for the details of how encoding, responding, and thinking are integrated is at the core of the ACT-R architecture and is an essential strength of the path-mapping theory. In this section we describe a story-mapping study that explores how mapping interacts with other cognitive processes. We show that low-level data such as eye-movement and typing data reveal the step-by-step nature of how people coordinate analogical mapping with other skills during a task. We then demonstrate a path-mapping model that captures human behavior even at this low level of analysis.

The story-mapping experiment involved mapping corresponding characters between two related stories such as those shown in Table 7. Each trial in the experiment comprised two phases: a study phase, where subjects studied a source story involving three characters and relations; and a mapping phase, where subjects read a similar target story and generated mappings between characters in the two stories. To examine behavior at a detailed level, we collected eye-movement and typing data as subjects performed the task. Researchers have utilized eye-movement data in domains such as reading (Just & Carpenter, 1984; Rayner, McConkie, & Zola, 1980) and mathematics (DeCorte, Verschaffel, & Pauwels, 1990) to understand better the step-by-step processes involved in these domains. This and other work have shown that eye movements can reveal a great deal about higher-level cognitive processes (Just & Carpenter, 1984; Rayner, 1995). Eye-movement data are especially useful because they represent actions at a fine grain size and do not require special training to produce informative data.

[[Insert Table 7 Approximately Here]]

Because the presentation of analogs is critical to eye movements in the task, we varied two factors to explore their effect on mapping behavior. First, we ran the experiment in two conditions to manipulate the availability of the source story when generating mappings for the target story. In the *one-story* condition, the source story was removed during the mapping phase, leaving subjects with only the target story on-screen. In the *two-story* condition, the source story remained on-screen during the mapping phase. We included both conditions in the hope that subjects' visual search of the source story in the two-story condition would help us understand subjects' mental search of the source story in the one-story condition. Second, we varied the presentation order of the analog relations to gain insight on how order affects the mapping process. Keane, Ledgeway, and Duff (1994) showed that relation ordering can significantly affect the time needed to complete a correct mapping. We wished to extend these results to examine the effects of order on the step-by-step process of encoding and responding during mapping.

The stories used in the experiment represent a necessary tradeoff between complexity and simplicity. On the one hand, we required that the stories be somewhat complex to elicit interesting behavior from subjects. On the other hard, accuracy limitations of the eye-tracking equipment forced us to keep the printed representation of the stories simple and uncluttered. Also, given that reading and parsing text was peripheral to the study, we strove to keep the text as close to the relation representation as possible while still allowing for standard syntax and grammar. The stories used were our best attempt to reconcile this tradeoff, and considering the overall success of the experiment, they seem to have been well-suited to the task.

Method

Subjects

Twenty-two Carnegie Mellon University students participated in the study. One subject in the one-story condition was omitted from analysis because we were unable to track his eye movements successfully. All analyses thus include 10 subjects in the one-story condition and 11 subjects in the two-story condition.

Materials

The story-mapping task involved completing the analogical mapping between characters in 16 pairs of related stories. Each story was based on three binary relations, where each relation described a person performing some action with some object. For a particular story pair, each relation in one story was similar to exactly one relation in the other story. In addition, all pairs of stories used the female names Jen, Kate, and Lori in one story and the male names John, Kurt, and Larry in the other story; names beginning with the letters *J*, *K*, and *L* were chosen because these letters lie adjacent to each other on a standard keyboard, facilitating subject responses. The story pair had a unique one-to-one mapping between similar relations, and thus a unique mapping between people in the stories. The relations for each story were translated to text as simply as possible and given a short introductory sentence to provide context for the story. Table 7 illustrates a sample story pair with the stories' respective relations and correct mapping.

The story pairs were varied according to an adjacency factor which determined whether the relations for corresponding names appeared in adjacent positions. The story pairs were also varied according to two other factors: whether characters whose names began with the same letter mapped to each other, and whether the stories contained a higher-order relation; with one minor exception,[4] neither factor produced any significant effects or interactions in our analyses and we do not discuss them further. The story-mapping task comprised two stories for each of the eight possible settings of these three factors, producing 16 story pairs in total.

The stories were presented to subjects on a Macintosh computer connected to an IScan (Cambridge, MA) eye-tracking system. The eye tracker calculates the centers of the subject's pupil and the reflection of an infrared source on the subject's eye. Based on these two points, it then computes the estimated point-of-regard look point on the screen. The eye tracker collects point-of-regard information at 120 Hz and sends it to the Macintosh computer, which records the eye-movement data along with typing data and current stimulus information.

Procedure

The experiment began with instructions on how the stimuli would appear for the subject's particular condition and how responses were to be entered. After being calibrated with the eye-tracking equipment, the subject solved a sample problem to become familiar with the task procedure. The subject then proceeded to solve the 16 story problems in a randomized order.

The procedure for a given trial was divided into two phases: a study phase and a mapping phase. Each phase began when the subject looked at a small rectangle in the upper-left quadrant of the screen. In the study phase, the source story appeared on the left-hand side of the screen and the subject could study the story for as much time as desired. When finished, the subject pressed a key to continue on to the mapping phase. In the mapping phase, the target story appeared on the right-hand side of the screen; in the two-story condition only, the source story also appeared on the left. Under each relation in the target story, the name of the character in the story would appear along with an editable text box. When the subject determined the source character who best corresponded with the name next to the selected box, the subject typed the first letter of that source character's name and the entire name appeared in the box. The cursor automatically moved to the next editable box; the subject could also move the cursor forward using the space bar or backward using the delete key. When finished, the subject pressed the enter key to end the trial.

Experiment Results and Discussion

Correctness

Correctness denotes whether the mappings generated by subjects corresponded to the correct mappings. The average correctness broken down by condition and adjacency is shown in Table 8. We studied the effects of condition and adjacency in a repeated-measures ANOVA (analysis of variance) with condition as the between-subjects factor and adjacency as the within-subject factor. The analysis revealed significant main effects of condition, *F*(1,19)=27.84, *MSE*=1.33, *p*<.001, and adjacency, *F*(1,19)=5.69, *MSE*=.30, *p*<.05. Their interaction was not significant, *p*>.2.

[[Insert Table 8 Approximately Here]]

The condition effect clearly shows that the ability to review information in the source analog to avoid recall failures serves as a great help in accurate analogical mapping. Subjects in the two-story condition perform extremely well with an overall correctness of 97%. Subjects in the one-story condition perform moderately well but make errors on approximately one-fifth of all trials, with an overall correctness of 79%.

The adjacency effect manifests that when corresponding objects appear in adjacent relations, subjects can more easily determine the correct mapping. One might have expected to observe an adjacency effect only in the two-story condition, but there was also a large difference in the one-story condition and the interaction of condition and adjacency was not significant. This surprising result indicates that adjacency affects analogical mapping not only when people access the source story visually but also when they access it mentally. This result provides evidence that subjects in both conditions perform a search process for an appropriate analog that begins with the adjacent object. For non-adjacent trials, the search process sometimes evaluates the adjacent object and decides that the correspondence is close enough to consider a match, producing the observed adjacency effect on correctness.

Trial Times

Trial time denotes the average total time required for subjects to complete both phases of a correctly solved trial. Trial time results are also included in Table 7. We analyzed subjects' trial times for each phase using, as for correctness, a repeated-measures ANOVA with the between-subjects factor of condition the within-subject factor of adjacency. For study time, the analysis showed a highly significant effect of condition, $F(1,19)=26.71$, $MSE=2.55e9$, $p<.001$. For mapping time, the analysis revealed a highly significant effect of adjacency, $F(1,19)=18.21$, $MSE=6.30e8$, $p<.001$. For both analyses, no other main effects or interactions were significant, $p>.1$.

As the ANOVAs indicate, the effect of condition on the total time lies in the study phase, where subjects require more than twice as much time in the one-story versus the two-story condition. This result is reasonable given that subjects knew the source story would disappear in the mapping phase, giving them incentive to memorize the source story instead of simply reading it. Mapping in the one-story condition takes slightly more time on average than in the two-story condition, but this difference is not significant.

Whereas the effect of condition lies solely in the study phase, the effect of adjacency lies solely in the mapping phase. As expected, adjacency has no effect on study time, since the target has not yet appeared and thus adjacency has no meaning. Adjacency has a clear effect on mapping time, however, with subjects requiring 2-6 additional seconds to complete non-adjacent story pairs. This result, recalling the correctness results, again suggests that for a particular target object, people generally begin searching for an analogous source object at the position adjacent to the target object. Once again, we observe the surprising lack of interaction of adjacency and correctness, such that adjacency aids mapping whether or not the source story is visible during mapping; in fact, the adjacency effect is larger in the one-story condition than in the two-story condition. Thus, even in their mental search for analogous source objects, people seem to start searching memory at the same position as the target object.

Eye Movements

Subjects' eye movements reveal a great deal about what information they utilize during the task and the time at which they utilize it. We begin by examining gaze counts for the sentences in the two conditions and two trial phases. Gazes have been frequently used as a sensible metric for analysis of eye movements (Just & Carpenter, 1984). To compute gazes, we first identified fixations using a two-state hidden Markov model (Salvucci & Anderson, 1998b) and assigned fixations to their closest visual target. The visual targets comprised the three story 1 relations (s1-1, s1-2, s1-3), the three story 2 relations (s2-1, s2-2, s2-3), the story introductions, and the response boxes. After determining fixation targets, we then collapsed consecutive fixations on the same target into single gazes. As a consequence, in the resulting gaze sequences, no two consecutive gazes can be attributed to the same target.

Subjects' gaze counts according to trial phase and condition are shown in Table 9. In the study phase, subjects in the one-story condition exhibit more gazes on story 1 than those in the two-story condition. The discrepancy suggests that two-story subjects merely read the story whereas one-story subjects read and memorize the story. For both conditions, more gazes appear on the middle target s1-2 rather than the outer targets s1-1 and s1-3. The pattern of gazes suggests that subjects read and memorize the story relations in an up-and-down manner, leading to more gazes on the middle target. As we will see, analysis of the first-order gaze transitions supports this conclusion.

[[Insert Table 9 Approximately Here]]

In the mapping phase, subjects exhibit a reading behavior much like that in the study phase: the larger number of gazes on the middle targets indicates a preference for up-and-down processing. In the two-story condition, subjects clearly make much use of the source story during the mapping process; in fact, the gaze counts for the two stories in this condition are quite comparable. We can presume that subjects begin by reading the target story (which they have not seen until this point) then proceed to map the target objects. For each target object, subjects search the source story for a similar relation and return to the target story to enter their response. An up-and-down search process would explain the gaze count distributions for the source story. In the one-story condition, subjects exhibit gaze counts very similar to those for the study phase. The fact that subjects move their eyes so often in this condition is somewhat surprising, given that presumably much of the analogizing occurs mentally. The high counts suggest the possibility that subjects move their eyes in accordance with their mental search — that is, as subjects mentally recall relations in the source story, their eyes follow the same pattern in looking at the target story. Similar behavior has been observed in empirical work that demonstrates how eye movements directly reflect cognitive processes (Just & Carpenter, 1976).

While gaze counts describe how often subjects looked at various visual targets, first-order gaze transitions describe the sequential aspects of their behavior. Gaze transitions represent the probability of making a transition from one screen target to another — that is, the probability of fixating a particular next target given that the current target is fixated. Table 10a shows the gaze transitions for the study phase, collapsed over both conditions; the values represent the probability of making a transition to the column target given the current position in the row target. The table reveals that, as we suggested earlier, subjects have a strong tendency to read up-and-down: transitions from the outer targets usually lead to the middle target, and transitions from the middle target can go in either direction. Note that the larger probability of going from s1-2 to s1-3 versus s1-1 indicates a slight bias toward the downward direction.

[[Insert Table 10 Approximately Here]]

Table 10b shows the transitions for the mapping phase in the one-story condition. Here we see a similar pattern of transitions — namely, a strong tendency to read up-and-down. Although the downward trend in transitions from s2-2 is slightly stronger than those from s1-2, overall the transitions closely reflect transitions in the study phase.

In the transitions for the mapping phase in the two-story condition, shown in Table 10c, the picture is more complex due to twice the number of visual targets. When examining the within-story transitions for the source and target stories (the upper-left and lower-right quadrants of the table), we again see the familiar transition pattern for up-and-down reading or searching behavior. The between-story transitions indicate a strong tendency to move to adjacent relations, shown on the diagonals in the lower-left and upper-right quadrants. We also see a bias toward moving to the middle target when making a transition between stories; this behavior could allow for more efficient encoding and/or searching of relations through parafoveal vision.

Typing

Typing behavior in the story-mapping task gives clues as to when subjects complete various steps in the mapping process. We first examine **key-time ratios**, defined as the time at which a key is pressed divided by the total time for the trial; we limit this examination to trials in which subjects hit exactly three keys before hitting the enter key to end the trial. Table 11 displays the key-time ratios for the first, second, and third keystrokes in the two conditions. On average, the first keystroke occurs approximately halfway through the trial. This result suggests that subjects interleave mapping and responding during the task, such that they map the relations and respond incrementally rather than mapping all relations before responding. The other two keystrokes occur later in the trial with the second keystroke occurring closer to the third than to the first.

[[Insert Table 11 Approximately Here]]

To test further whether subjects interleave mapping and responding, we explored how often subjects generate a keystroke before registering a gaze at the second or third story 2 relations (i.e., s2-2 or s2-3). This occurred in 18% of correct trials in the two-story condition and 7% of correct trials in the one-story condition. Thus, subjects sometimes map the first target analog before encoding the second and third analogs. Again, we see strong evidence that subjects interleave mapping and responding and can map stories incrementally.

Discussion

In examining the integration of analogical mapping and general problem solving, the story-mapping study raises two important issues for theories of analogical mapping. First, subjects need not follow a strict serial process of encoding analog relations, mapping analogs, and generating responses; subjects clearly have the ability to map the first story 2 object before even encoding the last two objects. This (perhaps surprising) result indicates that people can divide up analogs in useful ways and map the parts incrementally. This type of incremental processing has been observed at the level of eye movements in other domains, such as tests of analytic intelligence (Carpenter, Just, & Shell, 1990). While most theories have recognized the importance of incremental processing in analogical mapping (e.g., Burstein, ???; Forbus, Ferguson, & Gentner, 1994; Keane & Brayshaw, 1988), they limit incremental processing to mapping itself and do not address how people interleave mapping with encoding and responding. The few researchers that have addressed interleaving of mapping and problem solving (e.g., Carbonell, ???; Holland, Holyoak, Nisbett, & Thagard, 1986) have provided ...???. The path-mapping theory allows for models that can represent such interleaving and thus can account for incremental processing both in the mapping process itself and in the coordination of mapping with other cognitive skills.

The second issue raised by the story-mapping study involves the effects of relation ordering on analogical mapping. While previous studies (e.g., Keane, 1997; Keane, Ledgeway, & Duff, 1994) have shown that order can affect mapping time when both analogs are present, the story-mapping study reveals that order can also affect mapping when only the target analog is visible; in other words, relation ordering affects not only visual search of the source analog when visible but also mental search of the source analog when not visible. This result bolsters the argument made by researchers (e.g., Hummel & Holyoak, 1997) that analogical mapping and perception are closely related in terms of basic cognitive mechanisms.

The Story-Mapping Model

We now present a path-mapping model that captures subject behavior in the story-mapping task. As was the case for the illustrative models described earlier, our modeling effort aims at accounting for behavior at the level of real-world metrics — namely correctness, trial times, eye movements, and typing. The complexity and size of the model precludes an exact description of every detail of the model. Thus, we focus our exposition on the model's crucial components that produce various effects observed in the empirical data.[5]

We describe the story-mapping model in terms of its component subgoals and how the subgoals interact. Figure 4 illustrates the general subgoal structure for the model; each box contains the name of a goal and points to underlying subgoals called by that goal. The top-most goal *solve-problem* pushes two subgoals: *study-story*, which implements the study phase of a trial, and *map-stories*, which implements the mapping phase of a trial. The following subsections detail the models of the study phase and mapping phase, along with the visual-motor model and parameter estimation.

[[Insert Figure 4 Approximately Here]]

Study Phase Model

The study phase represented by the *study-story* goal has two components. First, the model pushes the *read-story* subgoal to read story 1 for the first time. This process includes reading the introductory sentence, reading the story from top to bottom, and reading the story from bottom to top. Second, the model pushes the *memorize-story* subgoal, which memorizes the story so that it can be recalled in later analogizing; in the two-story condition, the model has the option of skipping this memorization with some probability (estimated, .80), since the story remains visible in the mapping phase. In memorization, the model repeatedly calls the *review-story* subgoal for alternating directions such that it reviews the story top-to-bottom, then bottom-to-top, etc. The *review-story* subgoal reads the story in the given direction and then attempts to recall each relation in the story. If the model recalls every relation successfully, the *review-story* subgoal returns success, otherwise it returns failure. Once the *review-story* subgoal returns success to the *memorize-story* goal, the model chooses with some probability whether to stop or to continue memorizing (estimated, .76) in order to make success more likely in the mapping phase. When the model stops memorizing, the *study-story* goal is popped and the study phase is completed.

This model of the study phase has several interesting aspects intended to capture subject behavior. The model, both in reading and reviewing story 1, alternates direction to read top-down then bottom-up. The gaze counts and transitions observed in the study results support the fact that subjects also read in this manner. This reading behavior is quite rational in that it allows for more efficient processing of analog information than reading only top-down and having to return to the top after each reading. The model also depends on the activation of relation chunks to determine how often it reviews the story. As the model reads the story, the activation of the story relations increases with each reading, making them more likely to be recalled in future reviews. Finally, the choice of whether to continue memorizing after successful review models the decision to provide oneself with more practice knowing that the activation of relation chunks (and thus the probability of recalling them) will decrease as time passes in the trial.

Mapping Phase Model

The mapping phase represented by the *map-stories* goal also has two components. First, the model chooses whether to subgoal *read-story* for story 2 with some probability (estimated, .80) or to skip this reading. The model reads story 2 in the same fashion as story 1 — that is, it reads the introduction, then the story top-down, then the story bottom-up. Second, the model subgoals the *incremental-map* goal to perform the analogizing. For each relation in story 2, the *incremental-map* goal attends to the relation and subgoals the *search* goal to determine its analogous story 1 relation. When the *search* goal is popped, the model passes the resulting analog back to the *incremental-map* goal, which types out the answer and moves on to the next story 2 relation.

The search process represented by the *search* goal examines each story 1 relation, beginning with the adjacent relation, and attempts to evaluate whether this relation is a proper analog. This evaluation uses path mapping for the current story 1 object with the current story 1 role and story 2 role as constraints, thus producing either a successful mapping or a failure. If the evaluation fails, the search continues in alternating directions until an analog is found; in the one-story condition, at each step of the search, the model can give up the search with a given probability (estimated, .02). During search, visual attention follows current cognitive processing: In the two-story condition, the model attends to the story 1 relation currently being considered; in the one-story condition, the model attends to the story 2 position corresponding to the story 1 relation currently being considered.

This model of the mapping phase, like that of the study phase, captures observed aspects of subject behavior in the task. The choice of whether to read story 2 before analogizing models the fact that subjects sometimes read story 2 first and sometimes generate the first mapping before ever looking at the second and third story 2 relations. The fact that search begins with the adjacent position produces rational, efficient eye movements and helps to model adjacency effects. The reading behavior observed during search, like that for reading and reviewing stories, alternates direction for efficient search. The fact that visual attention mirrors the current locus of cognitive attention has been seen in studies of visual attention (Just & Carpenter, 1976) and helps explain the similarities between observed search behaviors in the two experimental conditions.

Visual-Motor Model

The story-mapping model assumes a very simple model of visual attention and motor skills. For visual attention, the model encodes two different components of the stories: the introduction and the story relations. We assume a longer time for introduction encoding (estimated, 3 sec) and a shorter time for relation encoding (estimated, 300 ms). Whenever the model reads a relation, the visual module adds the appropriate role chunk to declarative memory; if the chunk already exists, its activation increases with repeated reading. For motor skills, the model types the first letter of analog names and also types the enter key to end each phase. We also set the time taken to type one character (estimated, 300 ms) to a value similar to those in recent empirical studies (Salthouse, 1986).

Parameter Estimation

Like all models in this paper, the story-mapping model uses the parameter values shown in Table 3. In addition, the model contains eight task-specific estimated parameters. We have already mentioned seven such parameters in the previous subsections. The remaining parameter, production strength (estimated, 5), affects the latency with which chunks can be retrieved (see Equation 3). Table 12 summarizes the task-specific parameters and values.

[[Insert Table 12 Approximately Here]]

Model Results and Discussion

Overall, the model provides excellent qualitative and quantitative fits to all aspects of the observed data. Table 8 shows that the model reproduces the observed condition and adjacency effects on correctness, *R*>.99. The model performs better in the two-story condition because it is able to review the story 1 analogs during search. However, it still makes occasional errors in this condition when activation noise causes an inappropriate analog to be retrieved. The model also performs better for adjacent problems: Because analog search begins with the adjacent relation, the model sometimes evaluates this relation incorrectly as an appropriate analogous relation. Table 8 also includes the predicted study and mapping times for the model, *R*=.93. The model captures the large condition effect in study time because of the need to memorize the story in the one-story condition. In addition, the model reproduces the adjacency effect in mapping time.

Table 9 shows the model's predictions for gaze counts over the various areas, *R*=.94. In the study phase, the model shows increased study in the one-story condition, plus larger gaze counts for the center relation (s1-2). This result suggests that the prediction of reading in alternating directions corresponds very well to subject behavior. In the mapping phase, the model again predicts larger gaze counts for the center relations and fits the data quite well except for the underprediction of s2-2 gazes in the one-story condition.

Table 10 includes the model predictions for gaze transitions, *R*=.91. For the study phase (Table 10a), the model's prediction of alternating reading directions generates close fits, including the slight bias toward downward transitions from s1-2. For the mapping phase in the one-story condition (Table 10b), the model shows similar trends but with more transitions between s2-1 and s2-3, due to the up-and-down visual search of story 2 in this condition. For the mapping phase in the two-story condition (Table 10c), the model tends to overpredict more likely transitions and underpredict less likely transitions. However, it captures many aspects of the sequential scanning behavior, such as more transitions along the path of up-down reading and more transitions between adjacent relations across stories.

Table 11 shows the model's predicted key-time ratios for the first, second, and third keystrokes, *R*=.97. Like subjects, the model generates its first keystroke approximately halfway through the trial and its second and third keystrokes nearer the end of the trial. Also like subjects, it predicts no effect of condition on key-time ratios. In addition, the model fits observed behavior in terms of the percentage of trials in which the first keystroke occurs before attending to s2-2 or s2-3: In the two-story and one-story conditions, the model predicts percentages of 22% and 9% while subjects produced percentages of 18% and 7%, respectively. The model predicts a smaller percentage for the one-story condition because the model moves its visual attention around story 2 as it considers the different relations in story 1.

In summary, by integrating path-mapping representations and mechanisms with task-specific organizational knowledge, the story-mapping model provides excellent fits to observed data in terms of several informative real-world metrics. A significant part of our modeling and fitting effort concerns aspects of the task other than those directly related to analogical mapping. However, it is important to show that the analogical mapping process can be placed in the context of a full model of subjects performing a task. The data conclusively show that mapping is intertwined with other problem-solving skills and that mapping is not simply an atomic module that is called once to return a complete mapping. The critical aspect of our model is its ability to model both the incremental nature of analogical mapping and the effects of memory access and adjacency on mapping success.

General Discussion

Relation to Other Theories and Models

The path-mapping theory relates closely to existing theories and models with respect to two critical issues. One important issue for the path-mapping theory and other theories is the interleaving of analogy and problem solving. Although interleaving has not been an emphasis of many existing theories, some researchers have recognized the importance of this issue and have addressed the issue to some extent. Holland, Holyoak, Nisbett, & Thagard (1986) described an analogical inference mechanism in the context of a small production system. Carbonell (1986) used analogy to derive solution plans for problem solving. Keane, Kahney, & Brayshaw (1989) ???. **?? I have to make sure about these references. ??** The path-mapping theory significantly extends these ideas by incorporating a theory of analogical mapping into a general cognitive architecture, thus allowing models of analogy to interface smoothly with models of other general and task-specific problem-solving skills.

Another important aspect of the path-mapping theory that relates to other theories concerns the issue of incremental mapping. Keane and Brayshaw (1988) constructed IAM, the first computational implementation of an incremental mechanism. Forbus, Ferguson, and Gentner (1994) extended the SME model (Falkenhainer, Forbus, & Gentner, 1989) to produce incremental structure mappings between analogs. Hummel and Holyoak's (1997) LISA model also exhibits incremental behavior by performing capacity-limited, incremental constraint satisfaction. COPYCAT??? Like these models, the path-mapping mechanism allows for incremental analogical mapping. However, the path-mapping theory differs from other theories in that the method by which mapping is incrementally performed — that is, the order in which the individual objects and relations are mapped — is embodied in organizational knowledge and thus may depend on particular task demands.

Path Mapping and Other Components of Analogy

The path-mapping theory addresses the central component process of analogy, namely the process of analogical mapping. However, analogy comprises a number of other component processes, such as access (Gick & Holyoak, 1980), adaptation (Novick & Holyoak, 1991), and schema induction (Gick & Holyoak, 1983). While many theories and computational models of analogy have already begun to address this issue rigorously, we have not yet done so with the path-mapping theory. Nevertheless, we have ample reason to be confident that an extension of the theory to other analogy processes would arise naturally and parsimoniously. We now outline possible extensions that illustrate how the theory might account for these other processes.

One important component of the analogy process is the access of an appropriate analog for a given analog. Researchers have shown that access is an essential part of successful analogizing and can sometimes be very difficult (Duncker, 1945; Gick & Holyoak, 1980). In the path-mapping theory, the success of the activation-based path-mapping mechanism suggests that a similar mechanism should handle the access of appropriate analogs. We have assumed in the path-mapping representation that analogs are strictly divided into two domains, namely the source and target domains. However, by including a domain slot in the role representation, we could label role chunks by domain and retrieve an appropriate target domain for a given source domain: For a given source role, a model could retrieve an analogous target chunk from a different domain and use this domain as the target domain. This process would explain how semantic similarity can affect access (Holyoak & Koh, 1987; Ross, 1989) and why familiar relations are more likely to be accessed (Inagaki & Hatano, 1987).

Inference is another component that plays a crucial role in the analogy process. Inference deduces new knowledge about the analogs by extending known analog structures using the results of analogical mappings; for instance, if a person knows that the electron revolves around the nucleus, she may use knowledge about the solar system to infer that the nucleus should be heavier than the electron. While theories of analogy have different viewpoints on exactly how inference is performed, their inference mechanisms can be characterized as implementing a process of copying with substitution and generation, or CWSG (see Holyoak, Novick, & Melz, 1994; Markman, 1997). In essence, CWSG takes derived mappings and copies the representational structures from one analog to the other, substituting objects with their respective corresponding objects. This rather general algorithm can be constrained by the systematicity, importance or salience, and pragmatics or relevance of the various parts of the analogs (Markman, 1997).

The path-mapping theory can utilize the CWSG (or a similar) algorithm to perform analogical inference. Because the results (i.e., mappings) of the path-mapping process remain in declarative memory, they can easily be retrieved and utilized by some later process such as inference. In the simplest case, the inference mechanism would traverse each analog and copy over to the other analog (with substitution) any relations absent in the latter analog. However, such a mechanism would not exhibit the incremental nature of the path-mapping theory of analogical mapping — it could feasibly infer enormous conceptual structures for long periods of time with no interruption or interleaving. A more plausible inference mechanism would, like the path-mapping mechanism, infer only parts of the analogs and allow for interleaving of other cognitive processes. And like path mapping, such an inference mechanism would exhibit variability, learning, and errors as dictated by the ACT-R architecture.

A third component of analogy that arises naturally from the path-mapping theory is schema induction. Schema induction involves deducing general schematic structures from derived mappings that can be employed for future problems (Gick & Holyoak, 1983; Novick & Holyoak, 1991; Ross & Kennedy, 1990). As for inference, a path-mapping model can make use of existing mappings in declarative memory to deduce more general schemata. This type of schema induction has been employed by ACT-R models for transfer to closely related problems (Salvucci & Anderson, 1998a). With additional processing and possible re-representational strategies, this method could very feasibly be extended for transfer to distantly related problems. Again, although a simpler, non-interleaved approach to schema induction is possible, a better model would allow for incremental schema induction and interleaved cognitive processing.

Cognitive Architectures and Theories of Analogy

In this article we have focused on how, through instantiation in a general cognitive architecture, the path-mapping theory integrates analogical mapping and general problem solving. We have emphasized two major advantages to this approach to understanding analogical mapping and, more broadly, analogy. First, the theory allows for the inclusion of task-specific organizational knowledge that coordinates mapping and other cognitive skills. Second, the theory allows for direct comparison of model predictions to empirical data in terms of real-world quantitative metrics. These are by no means the only advantages to this approach, however. We now mention one other important issue, working memory constraints, that illustrates and stresses the usefulness of designing theories of analogy within cognitive architectures.

Early theories of analogy and analogical mapping (Gentner, 1983; Holyoak & Thagard, 1989) generally ignored the constraints that working memory pose on analogy. These theories postulated algorithms that would process implausibly large analogs just as easily as very simple ones. Recently, theories and computational models of analogy have begun to address working memory constraints more systematically. For instance, IAM (Keane, Ledgeway, & Duff, 1994) and incremental structure mapping (Forbus, Ferguson, & Gentner, 1994) embody algorithms whose incremental nature is intended to reduce working memory load. Also, LISA (Hummel & Holyoak, 1997) imposes these constraints by imposing capacity limits on the size of analog phase sets.

The path-mapping theory, and the ACT-R architecture in which it is situated, help better understand how working memory constraints affect analogical mapping. The ACT-R architecture has addressed working memory issues in a number of domains (e.g., Anderson & Lebiere, 1998; Anderson & Matessa, 1997) by positing how chunks can be learned and retrieved, how they become more active with repeated practice, and how they decay with lack of use. Because it is implemented in the ACT-R architecture, the path-mapping theory necessarily inherits all the working memory constraints present in its parent architecture. For instance, path-mapping models can fail to retrieve analogous roles if these roles have not been frequently used; the activation of the role chunks can decay enough that, upon attempted retrieval, the final activation would fall below threshold and retrieval would fail. As analogs become larger and more complex, these models require more effort to maintain chunk activations above threshold; such behavior has been observed in list memory tasks, where rehearsal of the memorized list becomes more difficult as list length increases (Anderson, Bothell, Lebiere, & Matessa, 1998). These and similar effects illustrate that working memory constraints often affect how mapping interacts with other task-specific skills, such as those needed to encode analogs.

Working memory constraints in the path-mapping theory also help to explain the nature of individual and developmental differences in analogy. Lovett, Reder, & Lebiere (in press) demonstrated that individuals systematically vary in terms of the amount of attention they can direct toward the current goal. By varying this attentional parameter, they showed that ACT-R can account for a number of individual difference phenomena in memory-based tasks. The fact that the path-mapping theory inherits these aspects from the ACT-R architecture implies that the analogical mapping process is also subject to these basic capacity effects. In addition, this attentional parameter could also explain, in part, the effects of age differences on mapping ability (Gentner & Toupin, 1986; Halford, 1992).

Overall, the development of theories of analogy within more general cognitive architectures is a promising approach with numerous advantages. By providing a computational framework for the integration of analogy and general problem solving, these architectures offer the potential to model analogical behavior at an unprecedented level of detail. Both analogy theories and cognitive architectures have much to gain in this development: Cognitive architectures allow analogy theories to incorporate task-specific knowledge and utilize cognitive mechanisms common to all problem solving, while analogy theories help constrain the space of possible models developed in the cognitive architectures.

References

Anderson, J. R., Bothell, D., Lebiere, C., & Matessa, M. (1998). An integrated theory of list memory. *Journal of Memory and Language*, *38*, 341-380.

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson, J. R., & Matessa, M. P. (1997). A production system theory of serial memory. *Psychological Review*, *104*, 728-748.

Anderson, J. R., & Thompson, R. (1989). Use of analogy in a production-system architecture. In S. Vosniadou & A. Ortony (Eds.), *Similarity and Analogical Reasoning*. Cambridge, England: Cambridge University Press.

Carpenter, P. A., Just, M. A., & Shell, P. (1990). What one intelligence test measures: A theoretical account of the processing in the Raven Progressive Matrices Test. *Psychological Review*, *97*, 404-431.

Catrambone, R. (1994). Improving examples to improve transfer to novel problems. *Memory & Cognition*, *22*, 606-615.

DeCorte, E., Verschaffel, L, & Pauwels, A. (1990). Influence of the semantic structure of word problems on second graders' eye movements. *Journal of Educational Psychology*, *82*, 359-365.

Duncker, K. (1945). On problem solving. *Psychological Monographs*, *58*, 270.

Falkenhainer, B., Forbus, K. D., & Gentner, D. (1989). Structure-mapping engine. *Artificial Intelligence*, 41, 1-63.

Forbus, K. D., Ferguson, R. W., & Gentner, D. (1994). Incremental structure mapping. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society* (pp. 313-318). Hillsdale, NJ: Lawrence Erlbaum Associates.

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

Gentner, D. (1989). The mechanisms of analogical learning. In S. Vosniadou & A. Ortony (Eds.), *Similarity and Analogical Reasoning*. Cambridge, England: Cambridge University Press.

Gentner, D., & Landers, R. (1985). Analogical reminding: A good match is hard to find. In *Proceedings of the International Conference on Systems, Man, and Cybernetics*. New York: Institute of Electrical and Electronics Engineers.

Gentner, D., & Markman, A. B. (1994). Structural alignment in comparison: No difference without similarity. *Psychological Science*, *5*, 152-158.

Gentner, D., & Markman, A. B. (1997). Structure mapping in analogy and similarity. *American Psychologist*, *52*, 45-56.

Gentner, D., & Toupin, C. (1986). Systematicity and surface similarity in the development of analogy. *Cognitive Science*, *10*, 277-300.

Gick, M. L., & Holyoak, K. J. (1980). Analogical problem solving. *Cognitive Psychology*, *12*, 306-355.

Gick, M. L., & Holyoak, K. J. (1983). Schema induction in analogical transfer. *Cognitive Psychology*, *15*, 1-38.

Grudin, J. (1980). Processes of verbal analogy solution. *Journal of Experimental Psychology: Human Perception and Performance*, *6*, 67-74.

Halford, G. S. (1992). Analogical reasoning and conceptual complexity in cognitive development. *Human Development*, *35*, 193-217.

Hofstadter, D. R., & Mitchell, M. (1994). An overview of the Copycat project. In K. J. Holyoak & J. A. Barnden (Eds.), *Advances in Connectionist and Neural Computation Theory: Vol. 2. Analogical Connections* (pp. 31-112). Hillsdale, NJ: Lawrence Erlbaum Associates.

Holland, J. H., Holyoak, K. J., Nisbett, R. E., & Thagard, P. (1986). *Induction: Processes of inference, learning, and discovery*. Cambridge, MA: MIT Press.

Holyoak, K. J., & Koh, K. (1987). Surface and structural similarity in analogical transfer. *Memory & Cognition*, *15*, 332-340.

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

Holyoak, K. J., & Thagard, P. R. (1995). *Mental leaps: Analogy in creative thought*. Cambridge, MA: MIT Press.

Hummel, J. E., & Holyoak, K. J. (1997). Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review*, *104*, 427-466.

Inagaki, K., & Hatano, G. (1987). Young children's spontaneous personification as analogy. *Child Development*, *58*, 1013-1020.

Just, M. A., & Carpenter, P. A. (1976). Eye fixations and cognitive processes. *Cognitive Psychology*, *8*, 441-480.

Just, M. A., & Carpenter, P. A. (1984). Using eye fixations to study reading comprehension. In D. E. Kieras & M. A. Just (Eds.), *New Methods in Reading Comprehension Research*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Just, M. A., & Carpenter, P. A. (1992). A capacity theory of comprehension: Individual differences in working memory. *Psychological Review*, *99*, 122-149.

Keane, M. T., & Brayshaw, M. (1988). The incremental analogy machine: A computational model of analogy. In D. Sleeman (Ed.), *Third European Working Session on Learning* (pp. 53-62). San Mateo, CA: Morgan Kaufmann.

Keane, M. T., Kahney, H., & Brayshaw, M. (1989). Simulating analogical mapping difficulties in recursion problems. In A. Cohn (Ed.), *Proceedings of the Annual Conference of the Society for the Study of Artificial Intelligence and Simulation of Behavior*. London: Pitman.

Keane, M. T., Ledgeway, T., & Duff, S. (1994). Constraints on analogical mapping: A comparison of three models. *Cognitive Science*, *18*, 387-438.

Kieras, D. E., & Meyer, D. E. (1997). A computational theory of executive cognitive processes and multiple-task performance: Part 1. Basic mechanisms. *Psychological Review*, *104*, 3-65.

Klahr, D., Langley, P., & Neches, R. (Eds.) (1987). *Production system models of learning and development*. Cambridge, MA: MIT Press.

Kokinov, B. N. (1998). Analogy is like cognition: Dynamic, emergent, and context-sensitive. In K. J. Holyoak, D. Gentner, & B. Kokinov (Eds.), *Advances in analogy research: Integration of theory and data from the cognitive, computational, and neural sciences*. Sofia, Bulgaria: NBU Press.

Kubose, T. T., Holyoak, K. J., & Hummel, J. E. (1998). Strategic use of working memory in analogical mapping. Manuscript under review.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Novick, L. R., & Holyoak, K. J. (1991). Mathematical problem solving by analogy. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *17*, 398-415.

Rayner, K. (1995). Eye movements and cognitive processes in reading, visual search, and scene perception. In J. M. Findlay, R. Walker, & R. W. Kentridge (Eds.), *Eye Movement Research: Mechanisms, Processes, and Applications* (pp. 3-21). New York: Elsevier Science Publishing.

Rayner, K., McConkie, G. W., & Zola, D. (1980). Integrating information across eye movements. *Cognitive Psychology*, *12*, 206-226.

Ross, B. H. (1987). This is like that: The use of earlier problems and the separation of similarity effects. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *13*, 629-639.

Ross, B. H. (1989). Distinguishing types of superficial similarities: Different effects on the access and use of earlier problems. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *15*, 456-468.

Salthouse, T. A. (1986). Perceptual, cognitive, and motoric aspects of transcription typing. *Psychological Bulletin*, *99*, 303-319.

Salvucci, D. D., & Anderson, J. R. (1998a). Analogy. In J. R. Anderson & C. Lebiere (Eds.), *The Atomic Components of Thought* (pp. 343-383). Hillsdale, NJ: Lawrence Erlbaum Associates.

Salvucci, D. D., & Anderson, J. R. (1998b). Tracing eye movement protocols with cognitive process models. In *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society* (pp. 923-928). Hillsdale, NJ: Lawrence Erlbaum Associates.

Spellman, B. A., & Holyoak, K. J. (1996). Pragmatics in analogical mapping. *Cognitive Psychology*, *31*, 307-346.

VanLehn, K. (1998). Analogy events: How examples are used during problem solving. *Cognitive Science*, *22*, 347-388.

Veale, T., & Keane, M. T. (1997). The competence of sub-optimal theories of structure mapping on hard analogies. In *Proceedings of the International Joint Conference on Artificial Intelligence*. San Mateo, CA: Morgan Kaufmann.

Acknowledgments

This work was supported in part by a National Science Foundation Graduate Fellowship awarded to Dario Salvucci and Office of Naval Research grant N00014-95-10223 awarded to John R. Anderson. We thank Art Markman, Mark Keane, and Clayton Lewis for their many insightful suggestions. We also thank Alex Petrov and the ACT-R research group for helpful questions and comments concerning this work.

Correspondence and requests for reprints should be sent to Dario Salvucci, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 15213. Email: < dario+@cs.cmu.edu >. ???

Footnotes

[1] Actually, the equation is an approximation to a more complex activation process described in Anderson and Lebiere, 1998. It is obtained in the ACT-R simulation by turning the Optimized Learning flag on.

[2] For reasons of space and clarity, Table 1 omits two productions: one that attempts to retrieve an analogous target role when the target role is specified as a constraint, and one that pops completed mappings.

[3] A description and full specification of every model presented here can be found from the Published Models link on the ACT-R home page, located at http://act.psy.cmu.edu/.

[4] The presence of higher-order relations produced a minor interaction with condition for study time, $F(1,19)=6.57$, $MSE=4.37e7$, $p<.05$, and mapping time, $F(1,19)=4.66$, $MSE=7.55e7$, $p<.05$. The small effect arose in the one-story condition, where subjects studied higher-order stories for slightly less time but took slightly more time to map these stories. However, when analyzing the total time computed as the sum of study and mapping times, this factor produced no significant effect or interaction, $p>.1$. Thus, we attribute the effect to a “pay now or pay later” strategy where some subjects studied higher-order stories less but then needed more time to solve those mappings.

[5] Like the illustrative models, the full story-mapping model is available from the Published Models link on the ACT-R home page.

Table 1.

Path mapping productions in pseudocode.

Return-Previous-Mapping

```
    IF    the goal is to map a source object
          and a mapping already exists for the source object
  THEN    pop the goal
```

Retrieve-Source-Role

```
    IF    the goal is to map a source object
          and a mapping does not exist for the source object
          and a source role can be retrieved
  THEN    set that source role
```

Reached-Source-Path-Root

```
    IF    the goal is to map a source object
          and a mapping does not exist for the source object
          and a source role cannot be retrieved
  THEN    note that the root node has been reached
```

Retrieve-Components

```
    IF    the goal is to map a source object
          and a source role has been set
          and this role contains a source relation, parent type, slot, and child type
  THEN    set the source relation, parent type, slot, and child type
```

Map-Source-Relation

```
      IF   the goal is to map a source object
           and the source relation has been set
           and its analogous target relation has not been set
    THEN   push a goal to map the source relation
```

Retrieve-Analog-At-Root

```
      IF   the goal is to map a source object
           and the root node has been reached
           and the parent type, slot, and child type are set
           and an analogous target role can be retrieved for these components
           and the target role contains a target object and target relation
    THEN   record the mapping from the source object to the target object
           and record the mapping from the source relation to the target relation
           and pop the goal
```

Retrieve-Analog-Below-Root

```
      IF   the goal is to map a source object
           and the target relation, parent type, slot, and child type have been set
           and an analogous target role can be retrieved for these components
           and the target role contains a target object
    THEN   record the mapping from the source object to the target object
           and pop the goal
```

Table 2.

Sample trace when mapping at-electron in the atom and solar system domains.

Cycle	Production	Description
0	<Initial State>	Mapping *ss-planet*
1	Retrieve-Source-Role	Set source role to *ss-revolver*
2	Retrieve-Components	Set source relation to *ss-revolves*, parent type to *revolves*, slot to *revolver*, and child type to *planet*
3	Map-Source-Relation	Mapping *ss-revolves*
4	Retrieve-Source-Role	Set source role to *ss-effect*
5	Retrieve-Components	Set source relation to *ss-causes*, parent type to *causes*, slot to *effect*, and child type to *revolves*
6	Map-Source-Relation	Mapping *ss-causes*
7	Reached-Source-Path-Root	Reached root relation *ss-causes*
8	Retrieve-Analog-At-Root	Retrieved analogous role *at-effect*
		Mapped *ss-causes* to *at-causes*
		Mapped *ss-revolves* to *at-revolves*
9	Retrieve-Analog-Below-Root	Retrieved analogous role *at-revolver*
		Mapped *ss-planet* to *at-electron*

Table 3.

Estimated and preset parameters for ACT-R path mapping module.

Type	Description	Value
Estimated	Similar-chunk mismatch	.1
Estimated	Dissimilar-chunk mismatch	1.7
Preset	Initial references	50
Preset	Activation learning decay rate	.5
Preset	Activation noise	.5

Table 4.

Phenomena for analogical mapping and how the models in this paper address them. The first seven phenomena are from Hummel and Holyoak (1997); the last two are from the authors. The models are abbreviated as follows: PP = probability-problem, SO = soap-opera, AM = attribute-mapping, Sh = sharing, CM = country-mapping, and KH = Karla-the-Hawk.

	Phenomenon	PP	SO	AM	Sh	CM	KH
1.	Isomorphism	X	X				
2.	Semantic similarity	X		X			
3.	Pragmatic centrality		X				
4.	Multiple possible mappings for one analogy		X				
5.	Correct initial correspondence facilitates finding subsequent mappings			X			
6.	Difficulty finding mapping for "unnatural" analogy problems			X			
7.	Possible to map predicates with different numbers of arguments				X		
8.	Generation of non-isomorphic mappings					X	
9.	Preference for many-to-one over one-to-many mappings					X	
10.	Ability to map complex analogs quickly						X
	Number of common parameters	3					
	Number of task-specific parameters	1	1	0	–	–	–
	Data points	5	10	4	–	–	–
	Correlation(s)	.93	.98	.97	–	–	–

Table 5.

Probability-problem relations (Ross, 1989), empirical data, and model predictions. Unparenthesized values represent empirical data; parenthesized values represent model predictions. Note: The empirical correctness value for the 0/– condition is the average of reported values from two experiments.

Problem	Condition	Relation	Correctness
Sample	All	assigned-to (cars, mechanics)	–
Test	+/+	assigned-to (cars, mechanics)	.60 (.63)
	+/–	assigned-to (mechanics, cars)	.42 (.33)
	0/+	assigned-to (computers, students)	.54 (.60)
	0/–	assigned-to (students, computers)	.39 (.38)
	0/0	assigned-to (students, counselors)	.48 (.55)

Table 6.

Relations used in the soap-opera experiment (Spellman & Holyoak, 1996).

Source	Target	
Bosses (Peter, Mary)	bosses (Nancy, John)	bosses (David, Lisa)
Loves (Peter, Mary)	loves (John, Nancy)	loves (Lisa, David)
Cheats (Peter, Bill)	cheats (Nancy, David)	
	cheats (Lisa, John)	

Table 7.

Story-mapping sample relations, stories, and correct mappings.

	Source	Target
Relations	rested-on (Larry, couch) prepared (Kurt, chicken) collected (John, ingredients)	readied (Kate, tent) gathered (Jen, firewood) slept-on (Lori, cot)
Stories	Kurt, John, and Larry were cooking dinner. Larry rested on the couch while Kurt prepared the chicken and John collected the ingredients.	Kate, Jen, and Lori were out camping. While Kate readied the tent and Jen gathered firewood, Lori slept her cot.
Mappings	Kurt → Kate John → Jen Larry → Lori	

Table 8.

Story-mapping correctness, study time (seconds), and mapping time by condition and adjacency. Unparenthesized values represent empirical data; parenthesized values represent model predictions.

Condition	Adjacent	Proportion Correct	Study Time	Mapping Time
one-story	yes	.85 (.84)	14.4 (12.0)	12.9 (13.1)
	no	.73 (.77)	14.4 (11.8)	18.3 (15.7)
two-story	yes	.99 (.94)	6.7 (7.0)	12.2 (13.1)
	no	.94 (.90)	6.6 (7.0)	14.5 (15.2)

Table 9.

Story-mapping gaze counts by trial phase and condition. Unparenthesized values represent empirical data; parenthesized values represent model predictions.

Phase	Condition	s1-1	s1-2	s1-3	s2-1	s2-2	s2-3
Study	one-story	4.5 (3.7)	6.3 (6.0)	3.8 (3.3)	-	-	-
	two-story	2.4 (2.4)	2.5 (2.9)	1.5 (1.5)	-	-	-
Map	one-story	-	-	-	3.1 (3.4)	6.2 (4.9)	4.6 (3.4)
	two-story	2.6 (2.1)	4.1 (3.2)	2.4 (2.1)	2.7 (2.8)	3.9 (3.6)	2.9 (2.8)

Table 10.

Story-mapping gaze transitions. Unparenthesized values represent empirical data; parenthesized values represent model predictions.

Table 10a.

Transition probabilities for the study phase in both conditions.

to →	s1-1	s1-2	s1-3
s1-1	-	.92 (1.00)	.08 (.00)
s1-2	.41 (.46)	-	.59 (.54)
s1-3	.18 (.00)	.82 (1.00)	-

Table 10b.

Transition probabilities for the mapping phase in the one-story condition.

to →	s2-1	s2-2	s2-3
s2-1	-	.89 (.87)	.11 (.13)
s2-2	.30 (.40)	-	.70 (.60)
s2-3	.15 (.17)	.85 (.83)	-

Table 10c.

Transition probabilities for the mapping phase in the two-story condition.

to →	s1-1	s1-2	s1-3	s2-1	s2-2	s2-3
s1-1	-	.39 (.52)	.10 (.00)	.22 (.25)	.21 (.12)	.09 (.12)
s1-2	.26 (.34)	-	.27 (.34)	.04 (.08)	.27 (.16)	.16 (.08)
s1-3	.06 (.00)	.33 (.52)	-	.05 (.12)	.21 (.12)	.35 (.25)
s2-1	.33 (.36)	.25 (.00)	.05 (.00)	-	.33 (.64)	.04 (.00)
s2-2	.08 (.00)	.41 (.28)	.10 (.00)	.14 (.22)	-	.26 (.50)
s2-3	.07 (.00)	.23 (.00)	.29 (.56)	.08 (.00)	.33 (.44)	-

Table 11.

Story-mapping key-time ratios by condition. Unparenthesized values represent empirical data; parenthesized values represent model predictions.

Condition	Key 1	Key 2	Key 3
one-story	.51 (.51)	.80 (.74)	.90 (.97)
two-story	.52 (.51)	.79 (.74)	.92 (.97)

Table 12.

Story-mapping model parameters and estimated values.

Parameter	Value
Probability of skipping memorization (two-story)	.80
Probability of continuing memorization after successful recall	.76
Probability of reading the target story	.80
Probability of giving up search (one-story)	.02
Time for introduction encoding	3 sec
Time for relation encoding	300 ms
Time for typing one character	300 ms
Production strength	5

Figure Captions

Figure 1. Sample analogs for the solar-system and atom domains.

Figure 2. Soap-opera empirical data (Spellman & Holyoak, 1996) and model predictions for the plot-extension and mapping tasks. The categories are defined according to whether mappings were consistent or inconsistent with pragmatic emphasis (CP or IP), consistent or inconsistent with the cheating relations (CC or IC), or otherwise (Other).

Figure 3. Attribute-mapping empirical data (Keane, Ledgeway, & Duff, 1994) and model predictions. Note: Because the singleton-last and none-similar conditions used essentially the same materials, these task times have been averaged into one condition.

Figure 4. Subgoal structure of the story-mapping model.

Figure 1.

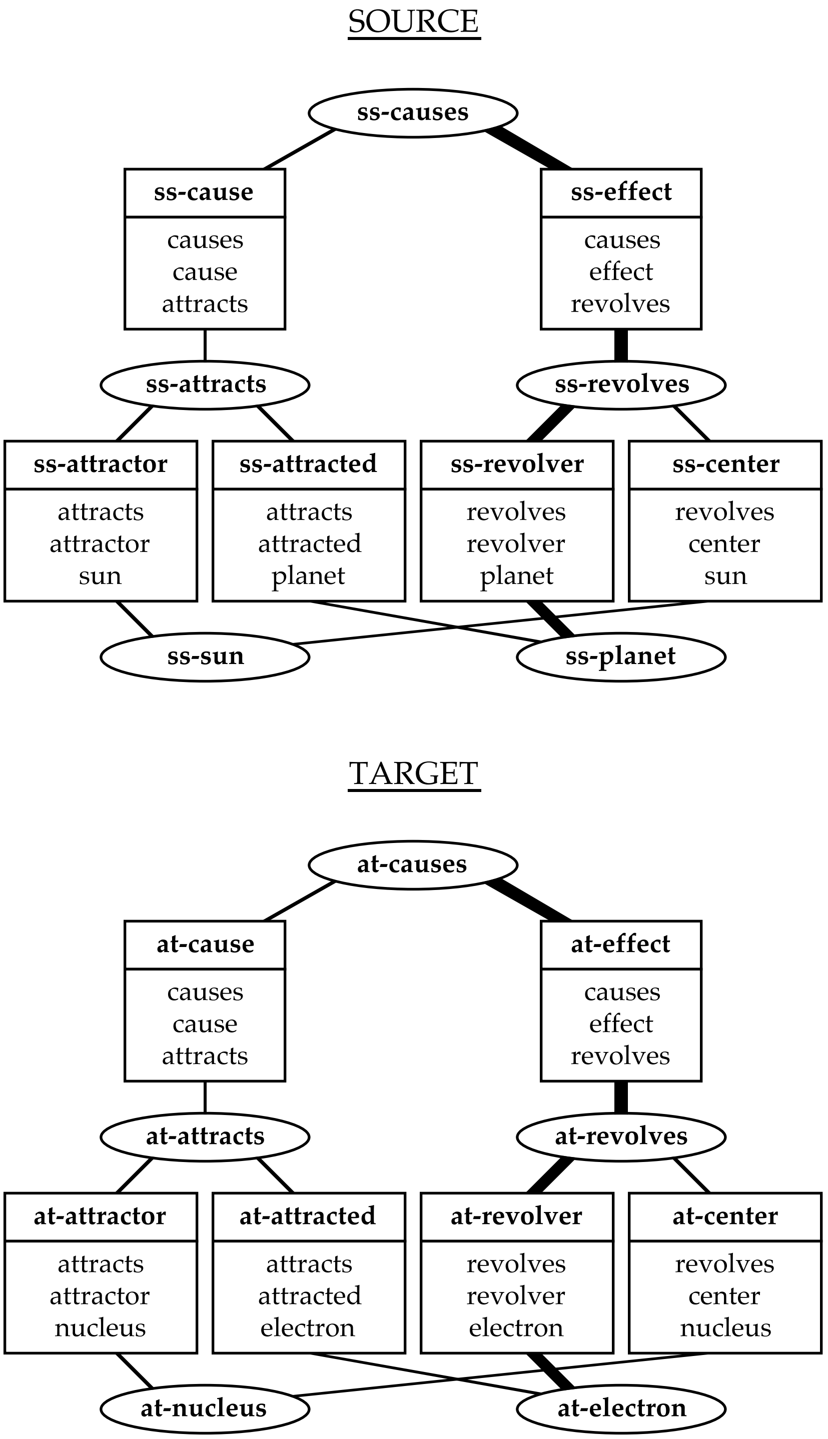

SOURCE
ss-causes
ss-cause
causes
cause
attracts
ss-effect
causes
effect
revolves
ss-attracts
ss-revolves
ss-attractor
attracts
attractor
sun
ss-attracted
attracts
attracted
planet
ss-revolver
revolves
revolver
planet
ss-center
revolves
center
sun
ss-sun
ss-planet
TARGET
at-causes
at-cause
causes
cause
attracts
at-effect
causes
effect
revolves
at-attracts
at-revolves
at-attractor
attracts
attractor
nucleus
at-attracted
attracts
attracted
electron
at-revolver
revolves
revolver
electron
at-center
revolves
center
nucleus
at-nucleus
at-electron

Figure 2.

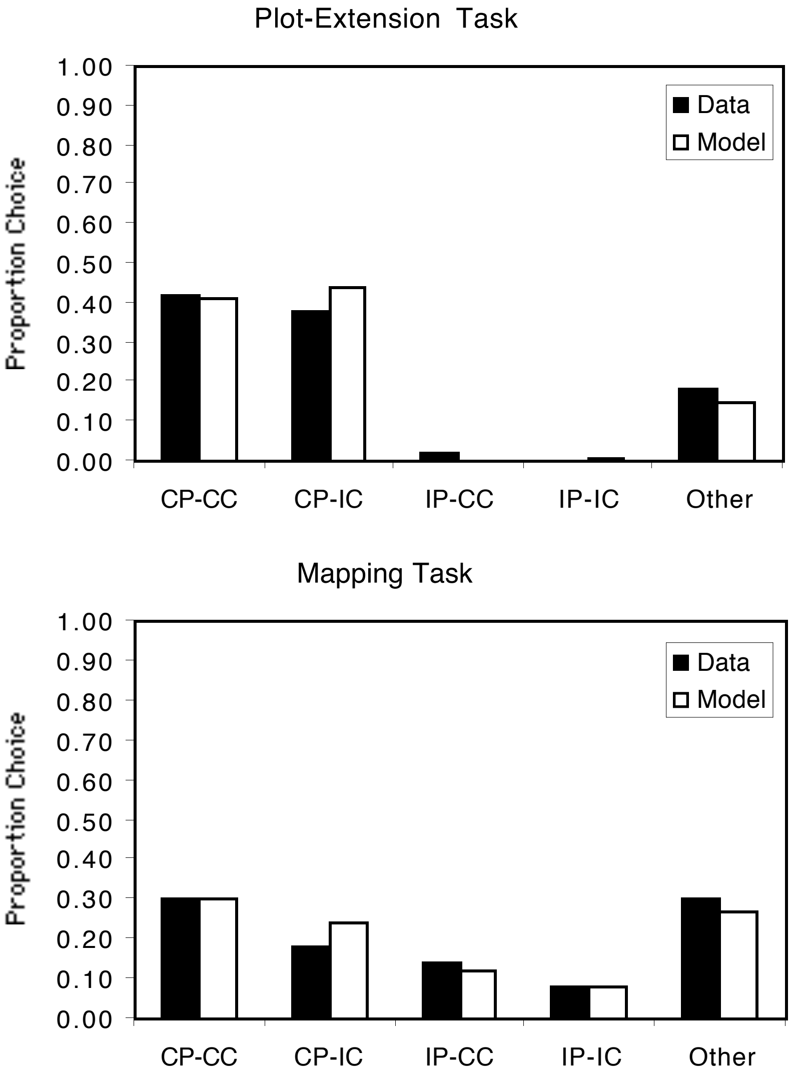

Plot-Extension Task
Proportion Choice
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
■ Data
□ Model
CP-CC
CP-IC
IP-CC
IP-IC
Other
Mapping Task
Proportion Choice
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
■ Data
□ Model
CP-CC
CP-IC
IP-CC
IP-IC
Other

Figure 3.

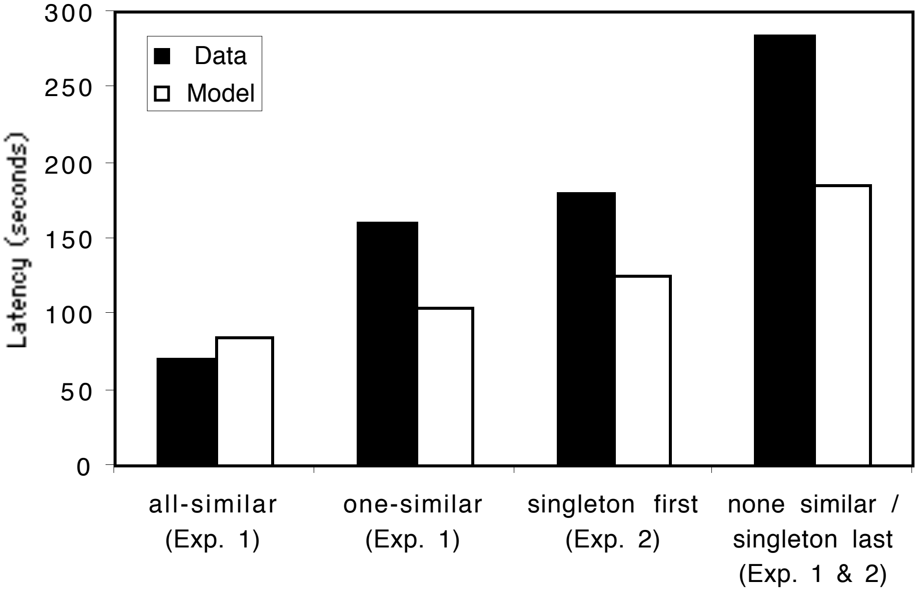
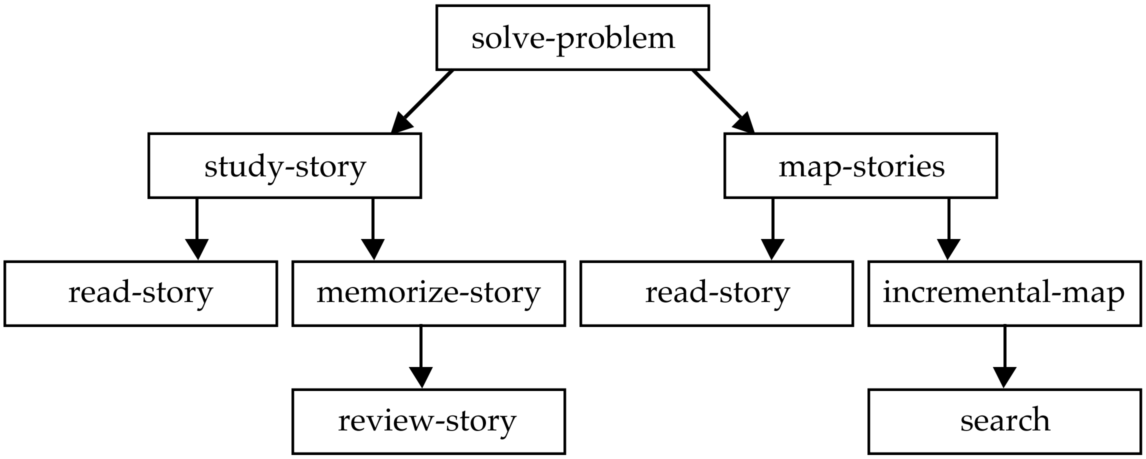

300
250
200
150
100
50
0
Latency (seconds)
■ Data
□ Model
all-similar (Exp. 1)
one-similar (Exp. 1)
singleton first (Exp. 2)
none similar / singleton last (Exp. 1 & 2)

Figure 4.


solve-problem
study-story
map-stories
read-story
memorize-story
read-story
incremental-map
review-story
search