Visuospatial Re-Representation in Analogical Reasoning

Jim Davies[*,1] and Ashok K. Goel[2]

[1]*Institute of Cognitive Science, Carleton University, Canada*

[2]*College of Computing, Georgia Institute of Technology, USA*

Abstract: Visual and spatial representations seem to play a significant role in analogy. In this paper, we describe a specific role of visual representations: two situations that appear dissimilar non-visuospatially may appear similar when re-represented visuospatially. We present a computational theory of analogy in which visuospatial re-representation enables analogical transfer in cases where there are ontological mismatches in the non-visuospatial representation. Realizing this theory in a computational model with specific data structures and algorithms first requires a computational model of visuospatial analogy, i.e., a model of analogy that only uses visuospatial knowledge. We have developed a computer program, called *Galatea*, which implements a core part of this model: it transfers problem-solving procedures between analogs that contain only visual and spatial knowledge. In this paper, we describe both how Galatea accomplishes analogical transfer using only visuospatial knowledge, and how it might be extended to support visuospatial re-representation of situations represented non-visually.

INTRODUCTION

A central issue in artificial intelligence research is that any solution to a problem has to start from what the program already knows: so, how is it possible to create novel solutions? There is ample evidence from cognitive science that, in humans, analogy plays an important role in finding solutions to novel problems, and that many of the most creative analogies involve cross-domain transfer [1-8], requiring finding similarity between semantically different ideas. But how can a program recognize similarity across domains and use it to arrive at a solution?

Clearly some kinds of abstraction processes are involved in transferring problem solutions across domains. Most of the literature on analogy considers abstraction processes using different kinds of non-perceptual, amodal (e.g., causal or functional) representations. We hypothesize that for some problems the reasoner may use abstraction processes that involve model re-representation, changing the representation from an amodal representation to a modal (e.g., visuospatial) representation. In these cases it is the similarity in the modal versions of these representations that enables analogical transfer.

To take an example, imagine that a reasoner is trying to figure out how to put batteries into a tape recorder by transferring a procedure from a source case in which film is put into a camera. Since *film* is a different entity from *battery*, and *tape-recorder* is a different entity from *camera*, analogical retrieval, mapping, and transfer will be hindered unless the reasoner can find some similarity between these pairs of entities. An *ontological mismatch* is where, given some representation language, entities are represented with different classes in the ontology.

One way the two situations in the above example are similar is that they visually resemble each other: the batteries and the film canister are both shaped like cylinders, and the tape recorder and the camera are shaped like rectangular prisms. In this example, the problem constraints pertain to the shapes of the objects involved. Thus the visuospatial similarity of the tape recorder and the camera (both may have cylindrical holes) is more relevant to the problem than, say, their functional similarity as recording devices, because their *shapes* have more to do with the placement of batteries and films than their *functions* do.

Our hypothesis is that turning a non-visuospatial representation into a visuospatial one (a process we call "visuospatial instantiation") is one mechanism for resolving ontological mismatches. When a reasoner encounters an ontological mismatch in non-visuospatial representations of the target and source analogs, it may dynamically create visuospatial representations of them. The problem-solving procedure can then be transferred between the generated visuospatial analogs. Then to translate the procedure and solution back to the non-visuospatial representation, the reasoner specifies the visuospatial representation back into a non-visuospatial form.

Note that a critical part of this process is transfer of a problem-solving procedure between two visuospatial analogs. Thus, if our hypothesis is correct, it should be possible to transfer problem-solving procedures from a source to a target problem using representations that are purely visuospatial. Therefore, the first core task in developing and evaluating our theory of multi-modal analogy is to develop and evaluate a computer program that can accomplish the transfer task using only visuospatial knowledge.

Indeed, although we outlined the case for visuospatial *re-representation* in analogy, in many situations analogical problem solving may use visuospatial representations from the outset. Problems in many design domains contain drawings, diagrams, animations, photographs, videos, etc. Instruc-

[*] Address correspondence to this author at the Institute of Cognitive Science, Carleton University, 22nd Floor, Dunton Tower, 1125 Colonel By Dr., Ottawa, Ontario, K1S 5B6, Canada; Tel: 613-520-2600, Ext. 1109;
E-mail: jim@jimdavies.org

1874-0618/08

tions for assembling complex artifacts are often presented to people in a completely diagrammatic form. Thus, establishing transfer of procedures using visuospatial knowledge alone not only supports our theory of multi-modal analogies but also is an important task by itself.

This paper, then, has two goals: (i) to describe our theory of visuospatial re-representation in multi-modal analogies and (ii) to describe our operational computer program called *Galatea* that instantiates the transfer task where the analogs are represented visuospatially. Galatea provides specific data structures for realizing a computational model of visuospatial analogy and specific algorithms for analogical transfer of visuospatial knowledge. We begin with a description of Galatea, and then show how it might be extended to realize visuospatial instantiation.

We will use Gick and Holyoak's classic fortress/tumor problem [9, 10] as a running example throughout this paper.[1]

Visuospatial Analogy: Transfer

The general's procedure for solving the fortress problem can be represented as a series of knowledge states and transformations between them. A knowledge state characterizes the steps in the procedure by specifying information about the elements in the state and relationships between them. A transformation takes in a knowledge state, changes its configuration in some way, and produces the next knowledge state in the sequence. The first knowledge state represents the initial description of the problem. Starting from the first knowledge state in the fortress story, the first transformation is to break up the army into smaller armies. This leads to the second knowledge state containing smaller armies. The second transformation is to move the armies to different roads, and so on.

This analysis, and our goal to model visuospatial analogy, leads to two constraints on the representations. First, knowledge states will be represented only visuospatially, i.e. as shapes, their sizes, locations, motions, and spatial relationships between each other. The transformations will be operations that change the visuospatial characteristics of the state. Further, two successive states will be connected by only one primitive transformation. Second, visuospatial information will be represented using symbolic, structured, descriptive representations. This is differentiated from depictive representations (e.g., bitmaps), where a depictive representation only "specifies the locations and values of points in space" [11]. The symbolic representation is appropriate for problem solving because it provides the standard benefits of discreteness, abstraction, ordering, and composition.

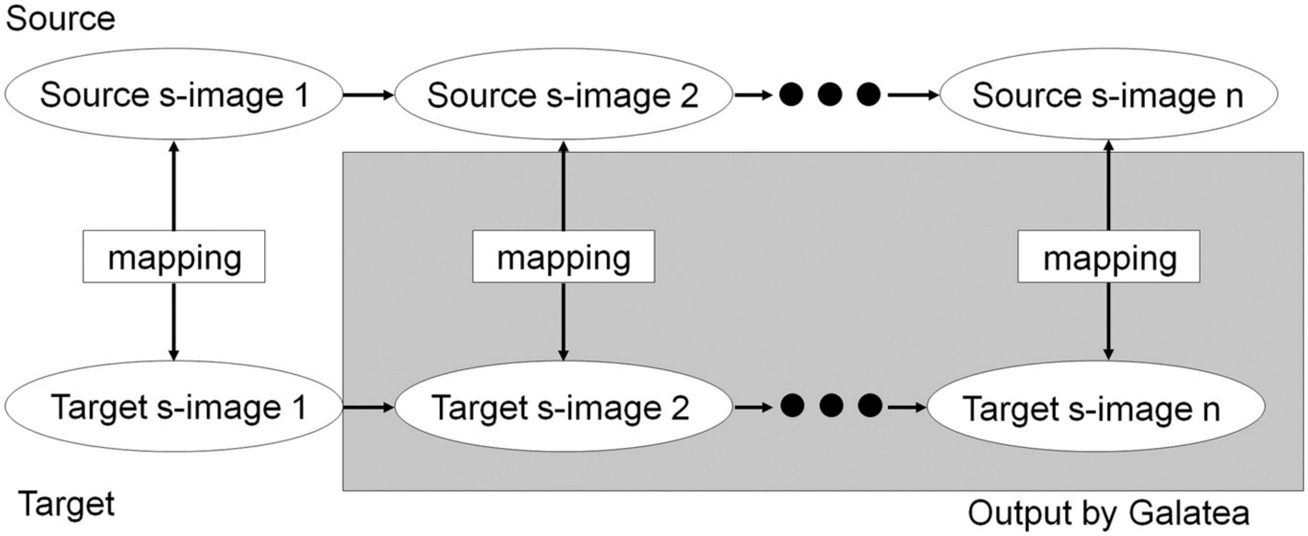

Fig. (1). Galatea's input and output in the abstract. The knowledge states in the source case are depicted as ovals along the top of the figure. The knowledge states are visually represented as s-images. Transformations between the states are depicted as arrows. The target problem is depicted as the first oval along the bottom. All things in the gray box are output by Galatea.

Galatea: A Computer Implementation

Galatea is an operational computer program written in LISP. It implements the analogical transfer of visuospatially-represented problem solving procedures. Fig. (**1**) illustrates Galatea's input and output in the abstract. Galatea's knowledge states are symbolic images or *s-images*. Galatea takes as input a source case, an initial target problem s-image, and an analogical mapping between the initial s-images of the source and target. The source case is a complete sequence of s-images and transformations representing the procedure that solves the source problem.

[1] Experimental participants read a story about a general who must overthrow a dictator in a fortress. His army is poised to attack along one of many roads leading to the fortress when the general finds that the roads are mined such that large groups passing will set them off. To solve the problem, the general breaks the army into smaller groups, and they take different roads simultaneously, arriving together at the fortress. Participants are then given a tumor problem, in which a tumor must be destroyed with a ray of radiation, but the ray will destroy healthy tissue on the way in, killing the patient. The analogous solution is to have several weaker rays simultaneously converging on the tumor [9, 10].

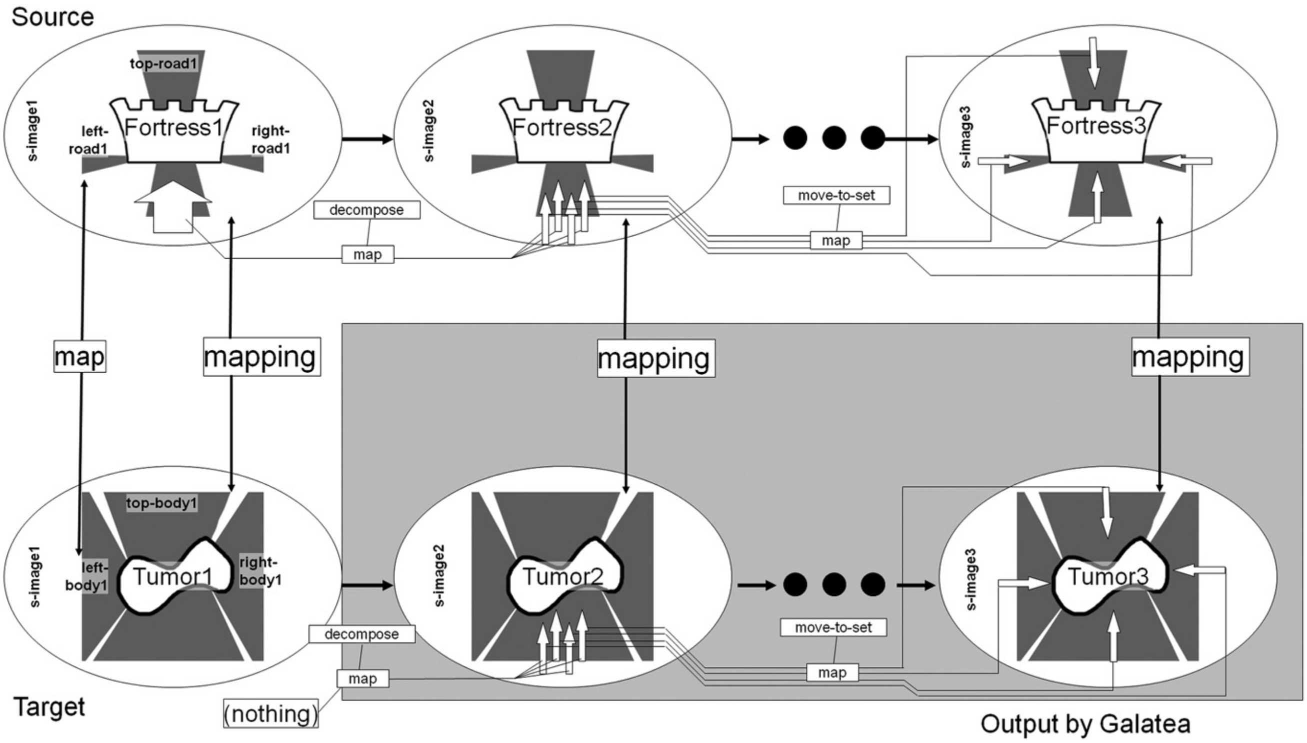

Fig. (2). Galatea's input and output for the fortress/tumor problem. The top series of s-images shows the visual representation of the solved fortress problem. The bottom series of s-images shows the target tumor problem. The bottom left s-image is the initial state of the tumor problem. The darkly shaded box shows Galatea's output.

Galatea transfers the visual transformations one at a time from the source to the target, creating new target s-images along the way, with new analogical mappings between the corresponding target and source s-images. Fig. (**2**) illustrates Galatea's input and output for the fortress/tumor problem.

Galatea is intended to be used for source and target problems with the following characteristic: the operators used to solve the problems must change the visuospatial characteristics of the situations. For example, changing the size of a door affects how the door looks, but changing who owns the door does not.[2] It presently works on three problems: The fortress/tumor problem, a case of scientific analogical reasoning by James Clerk Maxwell [12], and a cake/pizza problem in which a single pizza must be distributed among several people (as briefly described below).

Knowledge and Representation

Covlan (Cognitive Visual Language) provides an ontology of visual primitives. Table **1** shows Covlan's ontology of transformations.

Transformations implement normal graphics manipulations such as translation (*move-to-location*, *move-to-touch*, *move-above*, *move-to-right-of*, *move-to-left-of*, *move-below*), rotation (*rotate*), and scaling (*set-size*). In addition there are transformations for adding and removing elements from the

Table 1. Transformations in Covlan

Transformation Name	*Arguments*
move-to-location	*object, new-location*
move-to-touch	*object, object2, new-location*
move-above	*object, object2*
move-to-right-of	*object, object2*
move-below	*object, object2*
move-to-left-of	*object, object2*
move-in-front-of	*object, object2*
move-off-s-image	*object, location*
move-to-set	*object, object2*
rotate	*object, direction*
start-rotating	*object, direction*
stop-rotating	*object*
start-translation	*object, direction*
stop-translation	*object*
set-size	*object, new-size*
add-element	*object, location (optional)*
remove-element	*object*
decompose	*object, number-of-resultants, type*
scale	*object, new-size*

[2] Covlan, the visual language Galatea currently uses, does not address the visualization of abstract concepts like ownership, although such things could potentially be represented visually.

s-image (*add-element*, *remove-element*). Making topological changes of this kind to imagined physical systems has been shown in earlier work to be useful in problem solving [13, 14]. Like their counterparts in graphic manipulations, transformations take arguments to specify their behavior.

Certain transformations (*start-rotating*, *stop-rotating*, *start-translation*, *stop-translation*) are changes to the dynamic behavior of a single s-image. *Start-rotating* sets an element in motion, as one might spin a top. A square that has been affected by this transformation would not simply be rotated in the next s-image, but would be actively rotating in the next s-image. In contrast, *rotate* changes the orientation of an element once, as one might turn a chair a certain number of degrees to face a window. Such a transformation changes the position of an element between s-images.

Covlan's ontology of *primitive visual elements* (Table **2**) contains: *polygon*, *rectangle*, *triangle*, *ellipse*, *circle*, *arrow*, *line*, *point*, *curve*, and *text*. The elements are frame-like structures with slots that can hold values. For example, a *triangle* has a *location*, *size*, *height*, *width*, and *orientation*. All elements have a *location*, which is an absolute location on an s-image (e.g. *top*, *right*).

Table 2. Primitive Visual Elements in Covlan

Primitive Element Name	*Attributes*
polygon	*location, size*
rectangle	*location, size, height, width, orientation*
triangle	*location, size, height, width, orientation*
ellipse	*location, size, height, width, orientation*
circle	*location, size, height*
arrow	*location, length, start-point, end-point, thickness*
line	*location, length, end-point1, end-point2, thickness*
point	*location*
curve	*location, start-point, mid-point, end-point, thickness*
text	*location, length, letters*

In the fortress problem, the fortress is represented as a curve, the army as an *arrow* with *thickness* of *very-thick*. Likewise, in the tumor problem, the ray of radiation is represented as an *arrow* with *thickness* of *very-thick*, and the tumor is represented as a *curve* (Fig. **2**). Two s-images are generated during processing; the final generated s-image of the tumor problem is represented in Fig. (**3**).

In the fortress/tumor example, after the *decompose* transformation generates a number of smaller armies (by transforming a thick arrow into thinner arrows), they must be dispersed to the various roads, in various locations in the image. In a previous version of Galatea (Davies & Goel, 2001; Davies *et al*., 2005) each army arrow was *moved-to-location* individually to each road line. This solution was brittle because the number of roads to which the armies moved needed to match exactly the number of body areas the weaker rays moved to in the target.

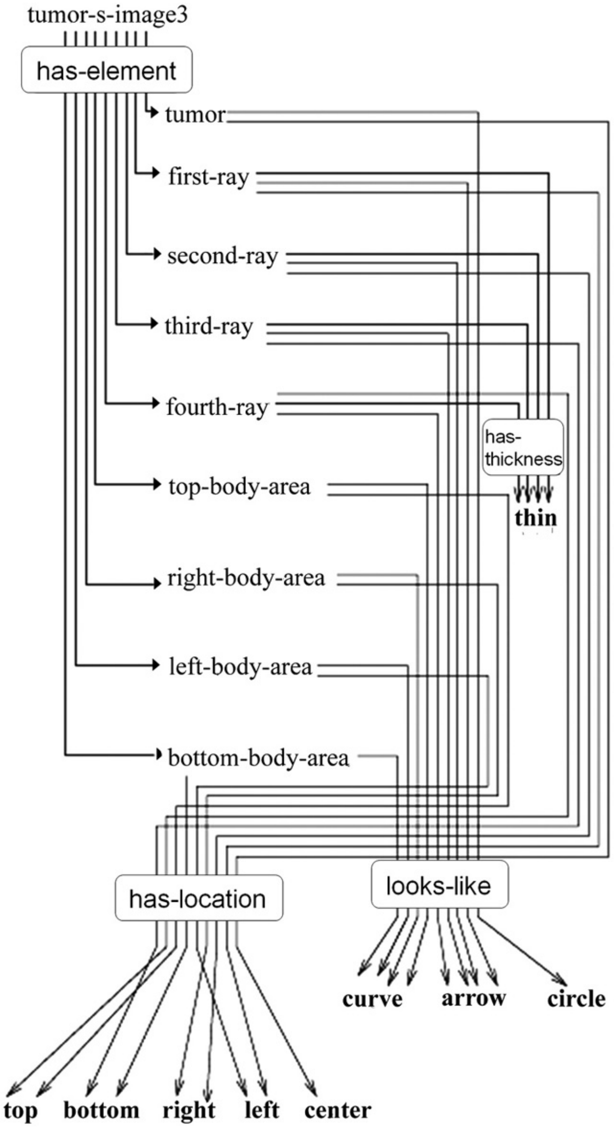

Fig. (3). A portion of the third s-image in the tumor series illustrated in Fig. (2). The third s-image in the tumor series represents the solution generated by Galatea as a final result of analogical transfer. The representation consists of propositions, indicated as labeled arrows connecting two elements. The *tumor-s-image3* is connected with a *has-element* relation to each primitive visual element in the s-image. The elements in the s-image each have a location and are connected to a primitive visual element type with a *looks-like* relation. Each ray, represented as an arrow, also has a *thickness*-in this s-image, *thin*. Each arrow also has start and end points, also with locations (not shown in the figure). The s-image is connected to the s-image before it with a *transform-connection*. Not shown in the figure are the maps that connect the elements of this s-image to the previous s-image, as well as the maps to the corresponding source s-image.

Galatea uses *sets* to address this problem. By grouping the armies, roads, rays, and body parts into their own sets, Galatea adapts the solution in the source analog to accommodate differing numbers of any of these elements. Rather than using the *move-to-location* transformation on each army, it uses *move-to-set* to the change the location of the *set* of armies. The argument to this function is a *set* of roads. The *move-to-set* function takes one set and distributes its members around the locations of another set.

Inference and Processing

Galatea focuses on the transfer and adaptation stage of analogy. In particular, it adapts and transfers each transformation in the source problem to the target.

A transformation, such as *decompose*, can be used to turn a primitive element into an arbitrary number of resultants, which is taken as an argument. An argument of a transformation can be an instance of one of three cases. First, the argument can be a literal, like the number *4* or the location *bottom*. Literals are transferred unchanged to the target.

Second, the argument could be a primitive element member of the source s-image. In this case, the transfer procedure operates on the analogous element in the target s-image. For example, in the fortress problem, the *soldier-paths* are moved to the *roads*. When *move-to-set* is transferred to the tumor problem, the argument *set-of-roads* is adapted to the analogous *set-of-body-areas*.

In the third case, the argument can be a function. Since this case does not occur in the fortress/tumor problem, we will use another example to describe it. Let us suppose that a reasoner needs to feed six people with one Sicilian slice sheet pizza. An analog in memory of cutting a sheet cake for four people is used to generate a solution. Transfer is still difficult because somehow the *four* in the cake analog must be adapted to the number *six* in the source analog. Knowing how many pieces into which to cut the cake or pizza depends on the number of people each problem. Some notion of count is needed. The use of *functions* as arguments to transformations helps address this problem. The cake analog is represented with a *function* that counts the number of people as its argument for the *decompose* transformation. This function has an argument of its own, namely the set of cake eaters, which during adaptation adapts into the set of pizza eaters. When the transformation is applied to the pizza, it counts the members of the set of people in the pizza problem (which results in six). *Decompose* produces six pieces of pizza in the next s-image.

Algorithm

1. **Identify the first s-images of the target and source cases.** These are the current source and target s-images.

2. **Identify the transformations and their associated arguments in the current s-image of the source case.** This step finds out how the source case gets from the current s-image to the next s-image. In the fortress/tumor example, the transformation is *decompose*, with *four* as the *number-of-resultants* argument (not shown).

3. **Identify the objects of the transformations.** The objects of the transformation are what objects, if any, the transformation acts upon. For the *decompose* transformation, the object is the *soldier-path1* (the thick arrow in the top left s-image in Fig. (**2**)).

4. **Identify the corresponding objects in the target problem.** *Ray1* (the thick arrow in the bottom left s-image) is the corresponding component of the source case's *soldier-path1*, as specified by the mapping between the current source and target s-images (not shown). A single object can be mapped to any number of other objects. If the object in question is mapped to more than one other object in the target, then the same transformation is applied to all of them.

5. **Apply the transformation with the arguments to the target problem component.** A new s-image is generated for the target problem (bottom middle) to record the effects of the transformation. The *decompose* transformation is applied to the *ray1*, with the argument *four*. The result can be seen in the bottom middle s-image in Fig. (**2**). The new rays are created for this s-image. Adaptation of the arguments can happen in three ways, as described above: If the argument is an element of the source s-image, then its analog is found. If the argument is a function, then the function is run (note that the function itself may have arguments which follow the same adaptation rules as transformation arguments). Otherwise the arguments are transferred literally.

6. **Map the original objects in the target to the new objects in the target.** A transform-connection and mapping are created between the target problem s-image and the new s-image (not shown). Maps are created between the corresponding objects. In this example it would mean a map between *ray1* in the left bottom s-image and the four rays in the second bottom s-image. A correspondence is also created between the *ray1* to the set of thinner rays. This system does not solve the mapping problem, but a mapping from the correspondences of the first s-image enables the mappings for the subsequent s-images to be automatically generated.

7. **Map the new objects of the target case to the corresponding objects in the source case.** Here the rays of the second target s-image are mapped to soldier paths in the second source s-image. This step is necessary for the later iterations (i.e., going on to another transformation and s-image). Otherwise the reasoner would have no way of knowing on which parts of the target s-image the later transformations would operate.

8. **Check to see if goal conditions are satisfied. If they are, exit, and the problem is solved.** If not, and there are further s-images in the source case, set the current s-image equal to the next s-image and go to step 1. If there are no further s-images, then exit and fail. Goal conditions are represented non-visuospatially [15].

Galatea shows that visuospatial knowledge alone, with no amodal knowledge, is sufficient for enabling analogical transfer, supporting a central hypothesis of our theory of visuospatial analogy. It suggests a computational model of analogy based on dynamic visuospatial knowledge that complements traditional models based on amodal knowledge.

Although Galatea does not yet address the issues of retrieval and mapping for visuospatial knowledge, other implemented computer programs have (e.g. [16, 17]). Thus, we confidently conjecture that visuospatial knowledge alone can enable the first three stages of analogy: retrieval, mapping, and transfer.

Multi-Modal Analogies: Re-Representation

Galatea addresses a core part of our theory of visuospatial analogy: analogical problem-solving transfer using only visuospatial knowledge. Our general theory, described in this section, suggests why and how visuospatial reasoning is useful even with cases whose representations need not be visuospatial. There is psychological evidence that humans make use of visuospatial information when doing problem solving in general [18-21], but the details of what makes visuospatial knowledge useful for analogy in natural and artificial reasoners is largely unknown.

Another way to frame this problem is that we do not know under what conditions it is useful for a reasoner to generate and process a visuospatial representation. Our work on Galatea suggests that one reason to use visuospatial representations is that ideas that are semantically distant with a non-visuospatial representation (e.g. a marching army and a ray of radiation) may be semantically closer with a visuospatial representation. Turning non-visuospatial representations into visuospatial ones (a process we call *visuospatial instantiation*) is one possible solution to the *ontological mismatch* problem.

The Ontological Mismatch Problem

One kind of ontological mismatch occurs when the symbols representing two similar things are different. In a non-visuospatial representation of the fortress/tumor problem, the ray and the army are represented with different symbols. Thus, without some notion of similarity between them, they cannot be aligned, which hinders analogical problem solving. Ontological mismatches can be encountered during analogical retrieval, mapping, or transfer.

Ontological mismatches can hinder retrieval of appropriate analogs. Psychological studies show that analogs are retrieved from memory based on surface similarity of the target analog to the retrieved source [22]. Similar ideas represented with different symbols will fail to *appear* similar to the reasoner.

Upon retrieval of an analog, the reasoner might have trouble with *mapping* ideas that need to be aligned, such as the *tumor* with the *fortress*, because they (or their relations with other components) are represented with different symbols.

Even if this mapping problem is overcome, the reasoner could still have a problem in transfer of the solution strategy. Suppose that the reasoner knows of a solved problem which involves breaking up an army into smaller groups. The army is represented as a group of constituent soldiers. The target problem involves a ray of radiation which must be turned into a number of rays, with less intensity. The ray might be represented as a form of energy, with a number associated with its intensity, a representation that serves some other task (e.g., so that numeric intensities can be added). Not having anticipated that the ray and army might need to be aligned, they could have been encoded with incompatible representations. The transformation applied to the army will not work on the ray because the representation of the ray, in this example, does not have constituent parts: breaking something into parts is different from dispersing energy.

The point of this analysis is to show that a reasonable non-visuospatial representation can fail for transfer in analogical problem solving. It is possible to represent this problem with no ontological mismatches (e.g. as in [23]), but ontological mismatches are bound to occur in any large knowledge base [24].

Resolution of Ontological Mismatches

In our theory, ontological mismatches encountered in non-visuospatial representations can be resolved by providing a level of visuospatial abstraction at which two symbols in question are identical or similar. This process of visuospatial instantiation offers a means for resolving ontological mismatches different from, say, using a type hierarchy. For example, the ray in the *fortress* and the *tumor* may not be under any same superordinate category. Representing them both as *curves*, however, shows a similarity between these distant concepts. This kind of visuospatial abstraction works especially well in conditions under which the visuospatial properties of the objects represented are related to the properties relevant to the current task (Fig. **4**).

For the retrieval task, the reasoner may visuospatially instantiate the elements in question, and find similarity through comparison of the visuospatial symbols. For example, the *ray* and the *soldier-path* might both visuospatially instantiate to *arrows*, (the shape, not the weapon) at which point the reasoner can identify the similarity by querying memory with *arrow*.

The mapping stage outputs alignments between elements of the source and target initial problem states. As in retrieval, visuospatial instantiation of the target and the source can help align the symbols. For example, visuospatial instantiation of the *tumor* and *fortress* as *curves* abstracts them to the same symbol, adding constraints that can be used to facilitate mapping.

In the transfer task, the transformations that connect the different knowledge states in the source are transferred to the target. The elements in the target problem that a transferred transformation affect are analogous to the elements that get affected in the source. Sometimes, however, there can be problems in transfer due to ontological mismatches. For example, as discussed above, trying to transfer the *break-up* transformation to the ray in the fortress/tumor problem will not work because the ray does not have constituent parts.

Visuospatial representations can be used as an intermediate level of abstraction in the transfer task as well. Let us suppose that in solving the fortress/tumor problem, both the army and the ray get visuospatially instantiated as a *line*. The *break-up* transformation, too, gets visuospatially instantiated as the *decompose* visual transformation. In the generated

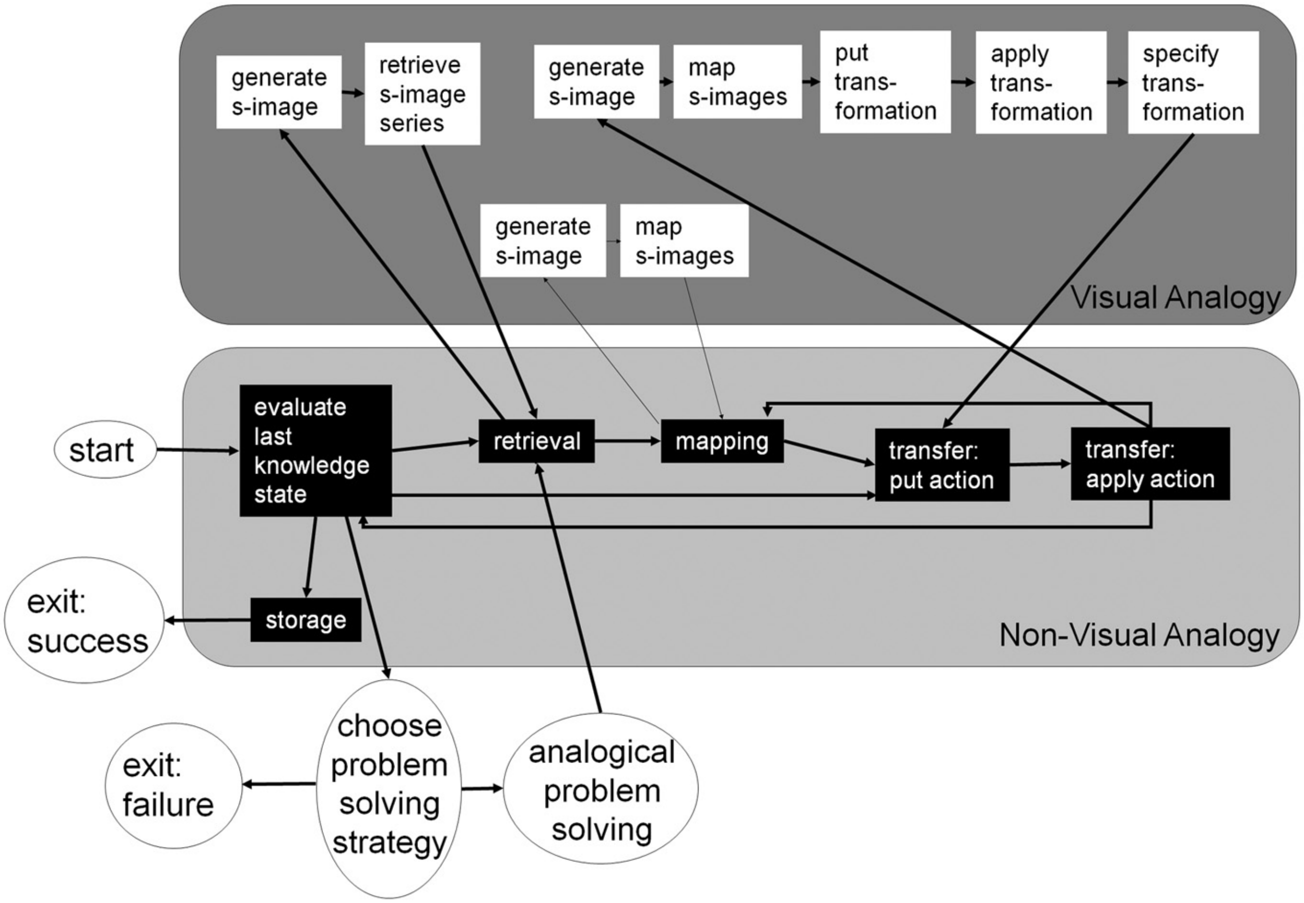

Fig. (4). The computational process for multi-modal analogies. First, the reasoner evaluates the last knowledge state generated in the target problem. If the goal conditions are satisfied, then the reasoner stores the target knowledge state sequence in memory and exits. Else it may elect to use analogy to address the problem. Retrieval can occur non-visually, but failing that, the reasoner may generate s-images for the target problem, and try to retrieve based on those (processes in the top shaded box are visual processes). When a source case is retrieved, the reasoner attempts to map the elements. Again, the visual nature of the generated s-images can be used to facilitate mapping. With a mapping in place, the reasoner may attempt to transfer the solution from the source to the target by transferring operators and applying them. If the transfer processes fail due to an ontological mismatch, the reasoner may use generated s-images and visual transformations to resolve it. The procedure transformations are re-specified back into non-visual operators. The final generated solution is evaluated in the non-visual representation.

visuospatial representation, the transfer of the transformation occurs without hindrance because *decompose* can apply equally well to both lines. This is exactly what happens in Galatea.

In Fig. (**5**), the top two ovals represent the first two knowledge states of a non-visuospatial representation of the source case in the fortress/tumor problem, connected with a *break-up* transformation. Also input is the initial state of the target problem and the analogical mapping between them. The grayed area is generated by the reasoner.

Decompose is a visual transformation that takes a visuospatial object and turns it into smaller objects of the same type. In this case, it turns the thick lines in the visuospatial target into thin lines. Unlike *break-up*, the *decompose* function transfers from the source to the target, because both *break-up* and *distribute* share the same visuospatial abstraction, decompose.

Now that the problem is solved in the visuospatial representation, it is re-specified back into the non-visuospatial representation. This can be done because *decompose* translates not only into *break-up* but also into *distribute*, which takes some intensity value and breaks it up into some number of elements with a weaker intensity.

Discussion

Our theory of visuospatial re-representation in analogical problem solving evolves from our earlier work, and shares two central themes with it: it provides a content account of analogy, and it uses abstractions for analogical transfer. The following discussion of related research is organized along these two themes.

Analogical problem solving pertains to transfer of complex relations from a source case to a target problem. Theories of analogical problem solving can be categorized as structure-based and content-based. Structure-Mapping Engine (SME) [22] and LISA [25] are two well-known structure-based theories of analogy. SME, for example, provides a uniform structure-based mechanism for analogical reasoning that is intended to work independently of any specific content account.

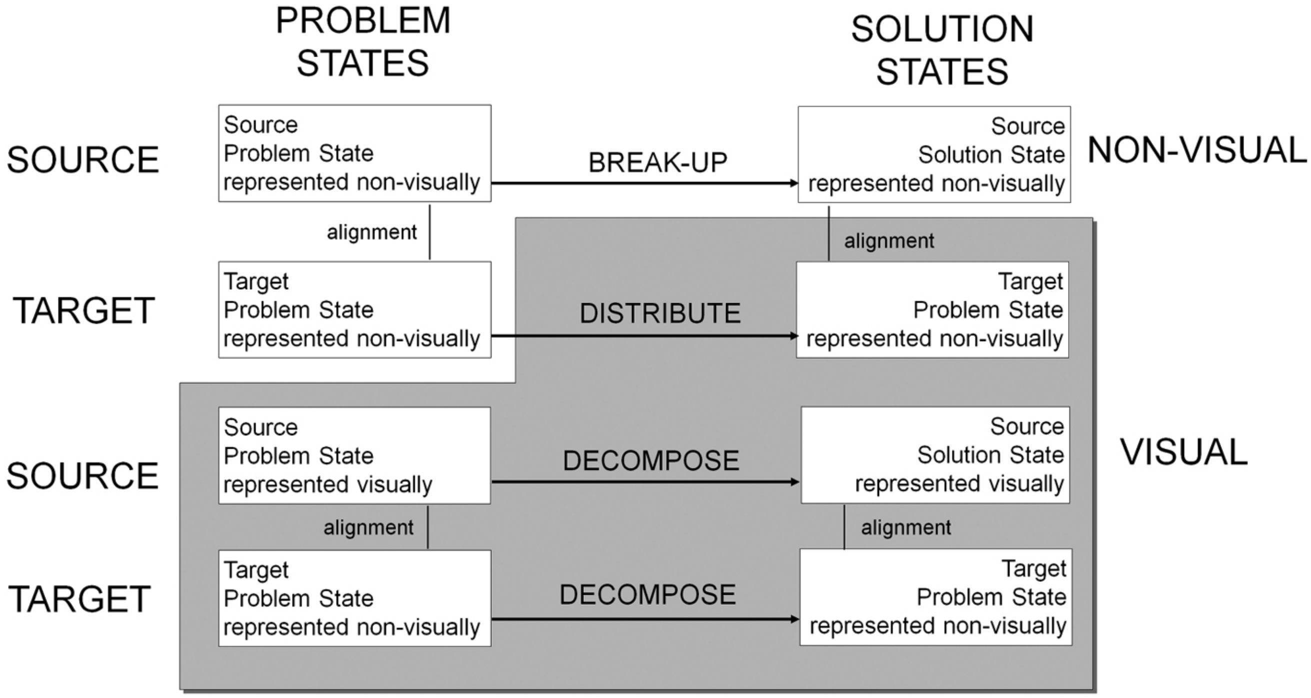
PROBLEM STATES
SOLUTION STATES
SOURCE
TARGET
SOURCE
TARGET
NON-VISUAL
VISUAL
Source Problem State represented non-visually
BREAK-UP
Source Solution State represented non-visually
alignment
alignment
Target Problem State represented non-visually
DISTRIBUTE
Target Problem State represented non-visually
Source Problem State represented visually
DECOMPOSE
Source Solution State represented visually
alignment
alignment
Target Problem State represented non-visually
DECOMPOSE
Target Problem State represented non-visually

Fig. (5). Visual re-representation in the transfer stage of analogical problem solving. When the reasoner cannot directly transfer the break-up transformation from the source case to the target problem, it creates a visual abstraction of the knowledge states and transformations. The non-visual break-up transformation instantiates to the visual transformation decompose. Transfer of decompose from the visual source to the visual target is now possible. After the transfer, the reasoner specifies the transferred visual transformation back into the appropriate non-visual transformation, distribute. Galatea implements the processing in the bottom half of this figure.

Analogical transfer in content-based theories, in contrast, is driven by content accounts of the domain. Winston [26] provides an early example of a content-based theory of analogical transfer. Our work too focuses on developing content-based theories of analogical problem solving. The IDeAL system [3, 4], for example, used structure-behavior-function (SBF) models for supporting analogical remindings, mappings, and transfer in the context of conceptual design. The ToRQUE system [13, 14] used SBF models for analogical remindings, transfer, and evaluation in the context of scientific problem solving. Qian and Gero [27] use similar Function-Behavior-Structure models to support analogy-based design. Galatea too is a content-based theory of analogical transfer. However, instead of functional and causal knowledge captured in SBF models, Galatea's uses visuospatial knowledge for the transfer task.

ANALOGY [28] and Letter Spirit [29] are two well-known models of visuospatial analogies. It solves multiple choice visuospatial analogy problems of the kind found on intelligence tests (e.g. A:B::C:?). It does this by describing how to turn A into B, and then testing which choice is a best fit for how C might be changed in a similar manner. Letter Spirit takes a stylized seed letter as input and outputs an entire font that has the same style. It does this by determining what letter is presented, determining how the components are drawn, and then drawing the same components of other letters the same way. The mappings between letters are already in the system: the vertical bar part of the letter *d* maps to the vertical bar in the letter *b*, for example. The seed letter may be interpreted as an *f* with the cross-bar suppressed. When the system makes a lower-case *t*, by analogy, it suppresses the crossbar.

Neither ANALOGY nor Letter Spirit transfers problem-solving procedures (ordered series of operations) as Galatea does. In contrast, one can see how Galatea might be applied to the font domain. The stylistic guidelines in Letter Spirit, such as “crossbar suppressed” are like the visual transformations in our theory: it would be a transformation of removing an element from the image, where that element was the crossbar and the image was a prototype letter *f*. Then the transformation could be applied to the other letters one by one. In this way our theory has more generality than Letter Spirit, which by design only works on alphabets.

The VAMP systems are analogical mappers [30]. VAMP.1 uses a hierarchically organized symbol/pixel representation. It superimposes two images, and reports which components have overlapping pixels. VAMP.2 represented images as agents with local knowledge. Mapping is done using ACME/ARCS [31]. The radiation problem mapping was one of the examples to which VAMP.2 was applied. Croft and Thagard [32] created a computational model DIVA which does analogical mapping using ACME. What it maps are three-dimensional visuospatial situations using representations in hierarchically organized scene graphs. Elements of the graph can be associated with behaviors, so it can represent dynamic systems. This system is a mapping system, and though it deals with the fortress/tumor problem, it does not transfer the solution procedure.

Though not an implemented computer program, the image-schema theory of Lakoff and Johnson [33, 34] says that humans use metaphors pertaining to their bodies to reason about external situations. Our theory is similar in that it uses perceptual abstraction to find similarity between ideas and to reason about external ideas. Our ideas differ in that their image-schemas are multi-sensory and based primarily on bodily action, where our theory, though it does not exclude such representations, focuses on visuospatial abstractions.

The perceptual symbol system theory [35] holds that all mental representations are perceptual in nature, so that all reasoning operates on perceptual symbols. On this view, what we have called the non-visuospatial or amodal representation is actually perceptual as well, and what we are calling the visuospatial level of representation is a more abstract perceptual representation.

MAGI [17] uses the structure-mapping theory to find examples of symmetry and repetition in a single image. JUXTA [36] uses MAGI in its processing of a diagram of two parts, and a representation of the caption. It outputs a description of what aligns with what, along with important and distracting differences. It models how humans understand repetition diagrams. Both systems are supported by the GeoRep [37] system, which generates visuospatial relations between elements. The visual primitives that describe a knowledge state in Galatea are similar to that of GeoRep.

Like Galatea, MAGI, JUXTA, and the VAMPs use visuospatial knowledge. But unlike Galatea their focus is on the creation of the mapping rather than on transfer of a solution procedure. MAGI's and our theory are compatible: a MAGI-like system might be used to create the mappings that our theory uses to transfer knowledge. The theory behind the VAMPs is incompatible because they use bitmaps, a different level of representation for the images.

The second central and consistent theme in our work is the use of abstractions to facilitate analogical transfer. In the IDeAL system, for example, analogical transfer was enabled by behavior-function abstractions of SBF models. IDeAL provided an account of the behavior-function abstractions in the form of generic teleological mechanisms and generic physical principles. The ToRQUE system similarly used generic structural transformations-abstractions to facilitate analogical reasoning.

Other models of analogical problem solving resolve the ontology mismatch problem in different ways. Some case-based reasoning theories appear to assume that memory is so massively populated and well organized and the retrieved case so similar to the target problem that ontological mismatches simply will not occur-if the source is so similar to the target that it need only be "tweaked" to get the desired solution, then there is simply no ontological mismatch. In contrast, Yarlett and Ramscar [38] specifically address the ontological mismatch problem in analogical reasoning. Their system takes two different symbols and evaluates their similarity using Latent Semantic Analysis [39], a database of correlations between all words representing their co-occurrence in text. The analogical mapper (in this case, SME) treats as identical any pair of symbols which correlate above a specified threshold. In our theory, objects and operations are treated as identical if they are visuospatially represented with the same visuospatial symbol.

Although the structure-mapping theory does not specifically address the ontological mismatch problem, as a structure-based theory it is applicable to both non-visuospatial and visuospatial representations. According to this theory, two ideas are considered similar if the idea's properties and the relations between it and surrounding elements are the same as the relations between another idea and its surrounding elements. For example, an electron is similar to a planet because both revolve around some body (a nucleus or star).

Galatea represents visuospatial abstractions symbolically. The system's design opts for higher-level visuospatial abstractions whenever possible. In future versions of Galatea, we plan to use multiple levels of visuospatial abstractions. We will also expand Galatea to computationally demonstrate how ontological mismatches in non-visuospatial representations can be aided by visuospatial instantiation and visuospatial reasoning according to the theory described in this paper.

CONCLUSION

This paper provides additional evidence that visuospatial analogical problem solving is computationally feasible on a machine. Purely visuospatial knowledge is sufficient for analogical problem solving for some cases. Of course, additional knowledge of goals, plans, causality and other non-visuospatial concepts would facilitate problem solving. This work also indicates that several representational features facilitate extraction and transfer of diagrammatic structure from source to target problems, including explicit representation of knowledge states in the form of symbolic images, a taxonomy of primitive visual transformations, and use of spatial aggregations.

According to Mednick [40] reasoners are creative because they can make semantic connections between dissimilar things. Solving the fortress/tumor problem by transferring the solution from the fortress problem to the tumor problem requires creativity of this kind. Our theory shows how this might occur: by changing representations to visuospatial and back again, natural and artificial reasoners can make connections among distant objects, strategies, and ideas.

ACKNOWLEDGEMENTS

This work has benefited from many discussions with Patrick Yaner and Nancy J. Nersessian.

REFERENCES

[1] N. J. Nersessian, "How do scientists think? capturing the dynamics of conceptual change in science." In R. N. Giere, Ed., *Cognitive models of science.* Minneapolis, Minneapolis: University of Minnesota Press, 1992.

[2] N. J. Nersessian, "Maxwell and "the method of physical analogy": Model-based reasoning, generic abstraction, and conceptual change." In D. Malament, Ed., *Essays in the history and philosophy of science and mathematics* (pp. 129-166). Lasalle, Il: Open Court, 2002.

[3] S. R. Bhatta, and A. K. Goel, "Design patterns: A computational theory of analogical design", in Proceedings of ijcai-97, workshop on using abstraction and reformulation in analogy, 1997.

[4] S. R. Bhatta, and A. K. Goel, "Learning generic mechanisms for innovative strategies in adaptive design." *J. Learn. Sci.*, vol. 6(4), pp. 367-396, 1997.

[5] P. Thagard, *Conceptual revolutions.* Princeton University Press, 1992.

[6] L. Darden, “Reasoning by analogy in scientific theory construction.” In Michalski, Ed., *Proc. Intl. Mach. Learn. Workshop*, pp. 32-40, 1983.

[7] D. Gentner, and A. L. Stevens, *Mental models*. Lawrence Erlbaum. 1983.

[8] M. Boden, The *creative mind: Myths and mechanisms*. London: Basic Books, 1990.

[9] M. L. Gick, and K. J. Holyoak, “Analogical problem solving”, *Cogni. Psychol.*, vol. 12, pp. 306-355, 1980.

[10] K. Duncker, “A qualitative (experimental and theoretical) study of productive thinking (solving of comprehensible problems)”, *J. Genet. Psychol.*, vol. 33, pp. 642-708, 1926.

[11] S. M. Kosslyn, *Image and brain: The resolution of the imagery debate.* Cambridge, MA: MIT Press, 1994.

[12] J. Davies, N. J. Nersessian, and A. K. Goel, “Visual models in analogical problem solving”, *Foundations of Science*, vol. 10(1), 2005. (Special issue on Model-Based Reasoning: Visual, Analogical, Simulative).

[13] T. W. Griffith, N. J. Nersessian, and A. K. Goel, “Function-follows-form transformations in scientific problem solving”, in Proceedings of the twenty-second annual conference of the cognitive science society, Lawrence Erlbaum, 2000.

[14] T. W. Griffith, N. J. Nersessian, and A. K. Goel, “The role of generic models in conceptual change”, in G. W. Cottrell, Ed., *Proceedings of the eighteenth annual conference of the cognitive science society,* 1996, pp. 312-317, San Diego, CA: Lawrence Erlbaum.

[15] J. Davies, and A. K. Goel, “Visual analogy in problem solving.” in B. Nebel, Ed., *Proceedings of the international joint conference for artificial intelligence 2001* Seattle, WA: Morgan Kaufmann Publishers, August 2001, pp. 377-382.

[16] P. W. Yaner, and A. K. Goel, “Retrieving 2-D line drawings by example”, in Proceedings of diagrams 2002, volume 2317 of lecture notes in artificial intelligence, pp. 97-99, 2002.

[17] R. W. Ferguson, “Magi: Analogy-based encoding using regularity and symmetry”, in A. Ram, and K. Eiselt, Eds., *Proceedings of the sixteenth annual conference of the cognitive science society*, Atlanta, GA: Lawrence Erlbaum Associates, 1994, pp. 283-288.

[18] J. Schrager, “Commonsense perception and the psychology of theory formation.” in J. Schrager and P. Langley, Eds., *Computational models of scientific discovery and theory formation*, pp. 437-470, San Mateo, California: Morgan Kaufman, 1990.

[19] M. J. Farah, “The neuropsychology of mental imagery: Converging evidence from brain-damaged and normal subjects.” in J. Stiles-Davis, M. Kritchevsky, and U. Bellugi, Eds., *Spatial cognition-brain bases and development,* pp. 33-59, Hillsdale, New Jersey: Erlbaum, 1988.

[20] H. Casakin, and G. Goldschmidt, “Expertise and the use of visual analogy: Implications for design education”, *Des. Stud.*, vol. 20, pp. 153-175, 1999.

[21] J. M. Monaghan, and J. Clement, “Use of computer simulation to develop mental simulations for understanding relative motion concepts”, *International J. Sci. Edu.,* vol. 21(9), pp. 921-944, 1999.

[22] B. Falkenhainer, K. D. Forbus, and D. Gentner, “The structure-mapping engine: Algorithm and examples”, *Artif. Intell.*, vol. 41, pp. 1-63, 1990.

[23] K. J. Holyoak, and P. Thagard, “A computational model of analogical problem solving”, in S. Vosniadou and A. Ortony, Eds., *Similarity and analogical reasoning*, pp. 242-266, Cambridge: Cambridge University Press, 1989.

[24] D. Lenat, and R. Guha, *Building large knowledge based systems: Representation and inference in the cyc project.* Reading, Massachusettes: Addison-Wesley Publishing, 1990.

[25] J. Hummel, and K. J. Holyoak, “Lisa: A computational model of analogical inference and schema induction”, in G. W. Cottrell, Ed., *Proceedings of the eighteenth annual conference of the cognitive science society,* pp. 352-357, San Diego, California: Lawrence Erlbaum Associates, 1996.

[26] P. H. Winston, “Learning and reasoning by analogy.” *Commun. ACM*, vol. 23(12), December 1980.

[27] L. Qian, and J. Gero, “Function-behaviour-structure paths and their role in analogy-based design.” *Artif. Intell. Eng.*, vol. 10, pp. 289-312, 1996.

[28] T. G. Evans, “A heuristic program to solve geometric analogy problems.” in M. Minsky, Ed., *Semantic information processing,* MIT Press, 1968. (ANALOGY program).

[29] G. McGraw, and D. R. Hofstadter, “Perception and creation of alphabetic style (Tech. Rep. No. SS-93-01)”, *AAAI,* (In Artificial Intelligence and Creativity: Papers from the 1993 Spring Symposium), 1993.

[30] P. Thagard, D. Gochfeld, and S. Hardy, “Visual analogical mapping”, in *Proc. 14th Annu. Conf. Cognit. Sci. Soc.*, pp. 522-527, Erlbaum, 1992.

[31] K. J. Holyoak, and P. Thagard, “The analogical mind”, *Am. Psychol.*, vol. 52(1), pp. 35-44, 1997.

[32] D. Croft, and P. Thagard, “Dynamic imagery: A computational model of motion and visual analogy”, in L. Magnani and N. J. Nersessian, Eds., *Model-based reasoning: Science, technology, & values*, pp. 259-274, New York: Kluwer Academic: Plenum Publishers, 2002.

[33] M. Johnson, *The body and the mind: The bodily basis of meaning, imagination, and reason,* University of Chicago Press, 1990.

[34] G. Lakoff, and M. Johnson, *Metaphors we live by,* Chicago: University of Chicago Press, 1980

[35] L. W. Barsalou, “Perceptual symbol systems”, *Behav. Brain Sci.*, vol. 22, pp. 577-609, 1999.

[36] R. W. Ferguson, and K. D. Forbus, “Telling juxtapositions: Using repetition and alignable difference in diagram understanding”, in K. Holyoak, D. Gentner, and B. Kokinov, Eds., *Adv. Analogy Res.,* pp. 109-117, New Bulgarian University, 1998.

[37] R. W. Ferguson, and K. D. Forbus, “Georep: A flexible tool for spatial representation of line drawings.” in *Proc.18th Natl. Conf. Artif. Intell.*, 2000.

[38] D. Yarlett, and M. Ramscar, “Structure-mapping theory and lexico-semantic information”, in L. R. Gleitman and A. K. Joshi, Eds., *Proceedings of the twenty-second annual conference of the cognitive science society*, pp. 571-576, Lawrence Erlbaum. 2000.

[39] T. K. Landaur, “Learning and representing verbal meaning: The latent semantic analysis theory.” *Curr. Direct. Psychol. Sci.*, vol. 7(5), pp. 161-164, 1998. (LSA: Latent Semantic Analysis).

[40] M. T. Mednick, “The associative basis of the creative process”, *Psychol. Rev.* vol. 3, pp. 220-232, 1962.

Received: December 18, 2007 Revised: March 11, 2008 Accepted: March 12, 2008